| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTree.DynamicUtilTreeNode.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/JTree.DropLocation.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JTree.EmptySelectionModel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JTree.DynamicUtilTreeNode.html)    [**NO FRAMES**](http://docs.google.com/JTree.DynamicUtilTreeNode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **javax.swing**

Class JTree.DynamicUtilTreeNode

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [MutableTreeNode](http://docs.google.com/javax/swing/tree/MutableTreeNode.html), [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html) **Enclosing class:**[JTree](http://docs.google.com/javax/swing/JTree.html)

public static class **JTree.DynamicUtilTreeNode**extends [DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html)

DynamicUtilTreeNode can wrap vectors/hashtables/arrays/strings and create the appropriate children tree nodes as necessary. It is dynamic in that it will only create the children as necessary.

**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).

| **Field Summary** | |
| --- | --- |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**childValue**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#childValue)            Value to create children with. |
| protected  boolean | [**hasChildren**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#hasChildren)            Does the this JTree have children? This property is currently not implemented. |
| protected  boolean | [**loadedChildren**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#loadedChildren)            Have the children been loaded yet? |

| **Fields inherited from class javax.swing.tree.**[**DefaultMutableTreeNode**](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) |
| --- |
| [allowsChildren](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#allowsChildren), [children](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#children), [EMPTY\_ENUMERATION](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#EMPTY_ENUMERATION), [parent](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#parent), [userObject](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#userObject) |

| **Constructor Summary** | |
| --- | --- |
| [**JTree.DynamicUtilTreeNode**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#JTree.DynamicUtilTreeNode(java.lang.Object,%20java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value, [Object](http://docs.google.com/java/lang/Object.html) children)            Creates a node with the specified object as its value and with the specified children. |

| **Method Summary** | |
| --- | --- |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html) | [**children**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#children())()            Subclassed to load the children, if necessary. |
| static void | [**createChildren**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#createChildren(javax.swing.tree.DefaultMutableTreeNode,%20java.lang.Object))([DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) parent, [Object](http://docs.google.com/java/lang/Object.html) children)            Adds to parent all the children in children. |
| [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html) | [**getChildAt**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#getChildAt(int))(int index)            Subclassed to load the children, if necessary. |
| int | [**getChildCount**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#getChildCount())()            Returns the number of child nodes. |
| boolean | [**isLeaf**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#isLeaf())()            Returns true if this node allows children. |
| protected  void | [**loadChildren**](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#loadChildren())()            Loads the children based on childValue. |

| **Methods inherited from class javax.swing.tree.**[**DefaultMutableTreeNode**](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) |
| --- |
| [add](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#add(javax.swing.tree.MutableTreeNode)), [breadthFirstEnumeration](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#breadthFirstEnumeration()), [clone](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#clone()), [depthFirstEnumeration](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#depthFirstEnumeration()), [getAllowsChildren](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getAllowsChildren()), [getChildAfter](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getChildAfter(javax.swing.tree.TreeNode)), [getChildBefore](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getChildBefore(javax.swing.tree.TreeNode)), [getDepth](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getDepth()), [getFirstChild](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getFirstChild()), [getFirstLeaf](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getFirstLeaf()), [getIndex](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getIndex(javax.swing.tree.TreeNode)), [getLastChild](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getLastChild()), [getLastLeaf](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getLastLeaf()), [getLeafCount](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getLeafCount()), [getLevel](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getLevel()), [getNextLeaf](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getNextLeaf()), [getNextNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getNextNode()), [getNextSibling](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getNextSibling()), [getParent](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getParent()), [getPath](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getPath()), [getPathToRoot](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getPathToRoot(javax.swing.tree.TreeNode,%20int)), [getPreviousLeaf](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getPreviousLeaf()), [getPreviousNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getPreviousNode()), [getPreviousSibling](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getPreviousSibling()), [getRoot](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getRoot()), [getSharedAncestor](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getSharedAncestor(javax.swing.tree.DefaultMutableTreeNode)), [getSiblingCount](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getSiblingCount()), [getUserObject](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getUserObject()), [getUserObjectPath](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getUserObjectPath()), [insert](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#insert(javax.swing.tree.MutableTreeNode,%20int)), [isNodeAncestor](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isNodeAncestor(javax.swing.tree.TreeNode)), [isNodeChild](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isNodeChild(javax.swing.tree.TreeNode)), [isNodeDescendant](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isNodeDescendant(javax.swing.tree.DefaultMutableTreeNode)), [isNodeRelated](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isNodeRelated(javax.swing.tree.DefaultMutableTreeNode)), [isNodeSibling](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isNodeSibling(javax.swing.tree.TreeNode)), [isRoot](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isRoot()), [pathFromAncestorEnumeration](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#pathFromAncestorEnumeration(javax.swing.tree.TreeNode)), [postorderEnumeration](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#postorderEnumeration()), [preorderEnumeration](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#preorderEnumeration()), [remove](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#remove(int)), [remove](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#remove(javax.swing.tree.MutableTreeNode)), [removeAllChildren](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#removeAllChildren()), [removeFromParent](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#removeFromParent()), [setAllowsChildren](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#setAllowsChildren(boolean)), [setParent](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#setParent(javax.swing.tree.MutableTreeNode)), [setUserObject](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#setUserObject(java.lang.Object)), [toString](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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()), [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** |
| --- |

### hasChildren

protected boolean **hasChildren**

Does the this JTree have children? This property is currently not implemented.

### childValue

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

Value to create children with.

### loadedChildren

protected boolean **loadedChildren**

Have the children been loaded yet?

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

### JTree.DynamicUtilTreeNode

public **JTree.DynamicUtilTreeNode**([Object](http://docs.google.com/java/lang/Object.html) value,  
 [Object](http://docs.google.com/java/lang/Object.html) children)

Creates a node with the specified object as its value and with the specified children. For the node to allow children, the children-object must be an array of objects, a Vector, or a Hashtable -- even if empty. Otherwise, the node is not allowed to have children.

**Parameters:**value - the Object that is the value for the new nodechildren - an array of Objects, a Vector, or a Hashtable used to create the child nodes; if any other object is specified, or if the value is null, then the node is not allowed to have children

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

### createChildren

public static void **createChildren**([DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) parent,  
 [Object](http://docs.google.com/java/lang/Object.html) children)

Adds to parent all the children in children. If children is an array or vector all of its elements are added is children, otherwise if children is a hashtable all the key/value pairs are added in the order Enumeration returns them.

### isLeaf

public boolean **isLeaf**()

Returns true if this node allows children. Whether the node allows children depends on how it was created.

**Specified by:**[isLeaf](http://docs.google.com/javax/swing/tree/TreeNode.html#isLeaf()) in interface [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html)**Overrides:**[isLeaf](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#isLeaf()) in class [DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) **Returns:**true if this node allows children, false otherwise**See Also:**[JTree.DynamicUtilTreeNode(java.lang.Object, java.lang.Object)](http://docs.google.com/javax/swing/JTree.DynamicUtilTreeNode.html#JTree.DynamicUtilTreeNode(java.lang.Object,%20java.lang.Object))

### getChildCount

public int **getChildCount**()

Returns the number of child nodes.

**Specified by:**[getChildCount](http://docs.google.com/javax/swing/tree/TreeNode.html#getChildCount()) in interface [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html)**Overrides:**[getChildCount](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getChildCount()) in class [DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) **Returns:**the number of child nodes

### loadChildren

protected void **loadChildren**()

Loads the children based on childValue. If childValue is a Vector or array each element is added as a child, if childValue is a Hashtable each key/value pair is added in the order that Enumeration returns the keys.

### getChildAt

public [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html) **getChildAt**(int index)

Subclassed to load the children, if necessary.

**Specified by:**[getChildAt](http://docs.google.com/javax/swing/tree/TreeNode.html#getChildAt(int)) in interface [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html)**Overrides:**[getChildAt](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#getChildAt(int)) in class [DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) **Parameters:**index - an index into this node's child array **Returns:**the TreeNode in this node's child array at the specified index

### children

public [Enumeration](http://docs.google.com/java/util/Enumeration.html) **children**()

Subclassed to load the children, if necessary.

**Specified by:**[children](http://docs.google.com/javax/swing/tree/TreeNode.html#children()) in interface [TreeNode](http://docs.google.com/javax/swing/tree/TreeNode.html)**Overrides:**[children](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html#children()) in class [DefaultMutableTreeNode](http://docs.google.com/javax/swing/tree/DefaultMutableTreeNode.html) **Returns:**an Enumeration of this node's children

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTree.DynamicUtilTreeNode.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/JTree.DropLocation.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JTree.EmptySelectionModel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JTree.DynamicUtilTreeNode.html)    [**NO FRAMES**](http://docs.google.com/JTree.DynamicUtilTreeNode.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

[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).