

---

# NSTreeNode Class Reference

Data Management: Data Types & Collections





Apple Inc.  
© 2006 Apple Computer, Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **NSTreeNode Class Reference 5**

---

- Overview 5
- Tasks 5
  - Creating Tree Nodes 5
  - Getting Information About a Node 5
  - Sorting the Subtree 6
- Class Methods 6
  - treeNodeWithRepresentedObject: 6
- Instance Methods 7
  - childNodes 7
  - descendantNodeAtIndexPath: 7
  - indexPath 7
  - initWithRepresentedObject: 8
  - isLeaf 8
  - mutableChildNodes 8
  - parentNode 9
  - representedObject 9
  - sortWithSortDescriptors:recursively: 10

---

## **Document Revision History 11**

---

## **Index 13**

---



# NSTreeNode Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Companion guide</b>	Cocoa Bindings Programming Topics
<b>Declared in</b>	NSTreeNode.h
<b>Related sample code</b>	AbstractTree DragNDropOutlineView SourceView

## Overview

`NSTreeNode` simplifies the creation and management of trees of objects. Each tree node represents a model object. A tree node with `nil` as its parent node is considered the root of the tree.

## Tasks

### Creating Tree Nodes

- + [treeNodeWithRepresentedObject:](#) (page 6)  
Creates and returns a tree node that represents the specified object.
- [initWithRepresentedObject:](#) (page 8)  
Initializes a newly allocated tree node that represents the specified object.

### Getting Information About a Node

- [representedObject](#) (page 9)  
Returns the object the tree node represents.
- [indexPath](#) (page 7)  
Returns the position of the receiver relative to its root parent.

- [isLeaf](#) (page 8)  
Returns whether the receiver is a leaf node.
- [childNodes](#) (page 7)  
Returns an array containing receiver's child nodes.
- [mutableChildNodes](#) (page 8)  
Returns a mutable array that provides read-write access to the receiver's child nodes.
- [descendantNodeAtIndexPath:](#) (page 7)  
Returns the receiver's descendent at the specified index path.
- [parentNode](#) (page 9)  
Returns the receiver's parent node.

## Sorting the Subtree

- [sortWithSortDescriptors:recursively:](#) (page 10)  
Sorts the receiver's subtree using the values of the represented objects with the specified sort descriptors.

## Class Methods

### **treeNodeWithRepresentedObject:**

Creates and returns a tree node that represents the specified object.

```
+ (id)treeNodeWithRepresentedObject:(id)modelObject
```

#### **Parameters**

*modelObject*

The object the tree node represents.

#### **Return Value**

An initialized tree node that represents *modelObject*.

#### **Availability**

Available in Mac OS X v10.5 and later.

#### **Related Sample Code**

DragNDropOutlineView

#### **Declared In**

NSTreeNode.h

## Instance Methods

### childNodes

Returns an array containing receiver's child nodes.

```
- (NSArray *)childNodes
```

#### Return Value

An array containing the receiver's child nodes.

#### Availability

Available in Mac OS X v10.5 and later.

#### Related Sample Code

DragNDropOutlineView

#### Declared In

NSTreeNode.h

### descendantNodeAtIndex:

Returns the receiver's descendent at the specified index path.

```
- (NSTreeNode *)descendantNodeAtIndex:(NSIndexPath *)indexPath
```

#### Parameters

*indexPath*

An index path specifying a descendent of the receiver.

#### Return Value

A tree node, or `nil` if the node does not exist.

#### Availability

Available in Mac OS X v10.5 and later.

#### Declared In

NSTreeNode.h

### indexPath

Returns the position of the receiver relative to its root parent.

```
- (NSIndexPath *)indexPath
```

#### Return Value

An index path that represents the receiver's position relative to the tree's root node.

#### Availability

Available in Mac OS X v10.5 and later.

**Related Sample Code**

AbstractTree

SourceView

**Declared In**

NSTreeNode.h

**initWithRepresentedObject:**

Initializes a newly allocated tree node that represents the specified object.

- (id)initWithRepresentedObject:(id)modelObject

**Parameters***modelObject*

The object the tree node represents.

**Return Value**An initialized tree node that represents *modelObject*.**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSTreeNode.h

**isLeaf**

Returns whether the receiver is a leaf node.

- (BOOL)isLeaf

**Return Value**

YES if the receiver is a leaf node (has no child nodes), otherwise NO.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

NSTreeNode.h

**mutableChildNodes**

Returns a mutable array that provides read-write access to the receiver's child nodes.

- (NSMutableArray \*)mutableChildNodes

**Return Value**

A mutable array that provides read-write access to the receiver's child nodes.

**Discussion**

Nodes that are inserted into this array have their parent nodes set to the receiver. Nodes that are removed from this array automatically have their parent node set to `nil`. The array that is returned is observable using key-value observing.

**Availability**

Available in Mac OS X v10.5 and later.

**Related Sample Code**

DragNDropOutlineView

**Declared In**

NSTreeNode.h

**parentNode**

Returns the receiver's parent node.

- (NSTreeNode \*)parentNode

**Return Value**

The receiver's parent node.

**Availability**

Available in Mac OS X v10.5 and later.

**Related Sample Code**

AbstractTree

DragNDropOutlineView

SourceView

**Declared In**

NSTreeNode.h

**representedObject**

Returns the object the tree node represents.

- (id)representedObject

**Return Value**

The object the tree node represents.

**Availability**

Available in Mac OS X v10.5 and later.

**Related Sample Code**

DragNDropOutlineView

**Declared In**

NSTreeNode.h

**sortWithSortDescriptors:recursively:**

Sorts the receiver's subtree using the values of the represented objects with the specified sort descriptors.

```
- (void)sortWithSortDescriptors:(NSArray *)sortDescriptors  
    recursively:(BOOL)recursively
```

**Parameters**

*sortDescriptors*

Array of sort descriptors specifying how to sort the represented objects.

*recursively*

A Boolean that specifies whether the child nodes should be sorted recursively.

**Discussion**

All the represented objects in the child nodes must be key-value coding compliant for the keys specified in the sort descriptors.

**Availability**

Available in Mac OS X v10.5 and later.

**Related Sample Code**

DragNDropOutlineView

**Declared In**

NSTreeNode.h

# Document Revision History

---

This table describes the changes to *NSTreeNode Class Reference*.

Date	Notes
2006-07-13	New document that describes the class used to manage a tree of objects.

**REVISION HISTORY**

Document Revision History

# Index

---

## C

---

childNodes **instance method** [7](#)

## D

---

descendantNodeAtIndexPath: **instance method** [7](#)

## I

---

indexPath **instance method** [7](#)  
initWithRepresentedObject: **instance method** [8](#)  
isLeaf **instance method** [8](#)

## M

---

mutableChildNodes **instance method** [8](#)

## P

---

parentNode **instance method** [9](#)

## R

---

representedObject **instance method** [9](#)

## S

---

sortWithSortDescriptors:recursively: **instance method** [10](#)

## T

---

treeNodeWithRepresentedObject: **class method** [6](#)