
NSAtomicStoreCacheNode Class Reference

[Cocoa](#) > [Data Management](#)



2006-10-09



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

NSAtomicStoreCacheNode Class Reference 5

- Overview 5
- Tasks 5
 - Designated Initializer 5
 - Node Data 5
- Instance Methods 6
 - initWithObjectID: 6
 - objectID 6
 - propertyCache 7
 - setPropertyCache: 7
 - setValue:forKey: 7
 - valueForKey: 8

Document Revision History 9

Index 11

NSAtomicStoreCacheNode Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/CoreData.framework
Availability	Available in Mac OS X v10.5 and later.
Companion guide	Core Data Programming Guide
Declared in	NSAtomicStoreCacheNode.h
Related sample code	Core Data HTML Store CustomAtomicStoreSubclass

Overview

`NSAtomicStoreCacheNode` is a concrete class to represent basic "nodes" in a Core Data atomic store.

A node represents a single record in a persistent store.

You can subclass `NSAtomicStoreCacheNode` to provide custom behavior.

Tasks

Designated Initializer

- [initWithObjectID:](#) (page 6)
Returns a cache node for the given managed object ID.

Node Data

- [objectID](#) (page 6)
Returns the managed object ID for the receiver.
- [propertyCache](#) (page 7)
Returns the property cache dictionary for the receiver.

- [setPropertyCache:](#) (page 7)
Sets the property cache dictionary for the receiver.
- [valueForKey:](#) (page 8)
Returns the value for a given key.
- [setValue:forKey:](#) (page 7)
Sets the value for the given key.

Instance Methods

initWithObjectID:

Returns a cache node for the given managed object ID.

```
- (id)initWithObjectID:(NSManagedObjectID *)moid
```

Parameters

moid

A managed object ID.

Return Value

A cache node for the given managed object ID, or `nil` if the node could not be initialized.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

Core Data HTML Store

CustomAtomicStoreSubclass

Declared In

NSAtomicStoreCacheNode.h

objectID

Returns the managed object ID for the receiver.

```
- (NSManagedObjectID*)objectID
```

Return Value

The managed object ID for the receiver.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

CustomAtomicStoreSubclass

Declared In

NSAtomicStoreCacheNode.h

propertyCache

Returns the property cache dictionary for the receiver.

```
- (NSMutableDictionary *)propertyCache
```

Return Value

The property cache dictionary for the receiver.

Discussion

This dictionary is used by `-valueForKey:` and `-setValue:forKey:` for property values. The default implementation returns nil unless the companion `-setPropertyCache:` method is invoked, or `-setValue:forKey:` is invoked on the cache node with non-nil property values.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSAtomicStoreCacheNode.h

setPropertyCache:

Sets the property cache dictionary for the receiver.

```
- (void)setPropertyCache:(NSMutableDictionary *)propertyCache
```

Parameters

propertyCache

The property cache dictionary for the receiver.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

CustomAtomicStoreSubclass

Declared In

NSAtomicStoreCacheNode.h

setValue:forKey:

Sets the value for the given key.

```
- (void)setValue:(id)value
    forKey:(NSString *)key
```

Parameters

value

The value for the property identified by *key*.

key

The name of a property.

Discussion

The default implementation forwards the request to the [propertyCache](#) (page 7) dictionary if *key* matches a property name of the entity for this cache node. If *key* does not represent a property, the standard `setValue:forKey:` implementation is used.

Availability

Available in Mac OS X v10.5 and later.

Related Sample Code

Core Data HTML Store

Declared In

NSAtomicStoreCacheNode.h

valueForKey:

Returns the value for a given key.

```
- (id)valueForKey:(NSString *)key
```

Parameters

key

The name of a property.

Return Value

The value for the property named *key*. For an attribute, the return value is an instance of an attribute type supported by Core Data (see [NSAttributeDescription](#)); for a to-one relationship, the return value must be another cache node instance; for a to-many relationship, the return value must be a collection of the related cache nodes.

Discussion

The default implementation forwards the request to the [propertyCache](#) (page 7) dictionary if *key* matches a property name of the entity for the cache node. If *key* does not represent a property, the standard `valueForKey:` implementation is used.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSAtomicStoreCacheNode.h

Document Revision History

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

Date	Notes
2006-10-09	New document that describes the Core Data class used to represent a node in an atomic persistent store.

REVISION HISTORY

Document Revision History

Index

I

initWithObjectID: [instance method 6](#)

O

objectID [instance method 6](#)

P

propertyCache [instance method 7](#)

S

setPropertyCache: [instance method 7](#)

setValue:forKey: [instance method 7](#)

V

valueForKey: [instance method 8](#)