
NSEntityMapping Class Reference

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





Apple Inc.
© 2007 Apple 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, Mac OS, and Xcode 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

NSEntityMapping Class Reference 5

Overview	5
Tasks	5
Managing Source Information	5
Managing Destination Information	6
Managing Mapping Information	6
Instance Methods	7
attributeMappings	7
destinationEntityName	7
destinationEntityVersionHash	8
entityMigrationPolicyClassName	8
mappingType	8
name	9
relationshipMappings	9
setAttributeMappings:	10
setDestinationEntityName:	10
setDestinationEntityVersionHash:	10
setEntityMigrationPolicyClassName:	11
setMappingType:	11
setName:	12
setRelationshipMappings:	12
setSourceEntityName:	12
setSourceEntityVersionHash:	13
setSourceExpression:	13
setUserInfo:	13
sourceEntityName	14
sourceEntityVersionHash	14
sourceExpression	15
userInfo	15
Constants	16
Entity Mapping Types	16
NSEntityMappingType	17

Document Revision History 19

Index 21

NSEntityMapping 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 Model Versioning and Data Migration Programming Guide
Declared in	NSEntityMapping.h

Overview

Instances of `NSEntityMapping` specify how to map an entity from a source to a destination managed object model.

Tasks

Managing Source Information

- [sourceEntityName](#) (page 14)
Returns the source entity name for the receiver.
- [setSourceEntityName:](#) (page 12)
Sets the source entity name for the receiver.
- [sourceEntityVersionHash](#) (page 14)
Returns the version hash for the source entity for the receiver.
- [setSourceEntityVersionHash:](#) (page 13)
Sets the version hash for the source entity for the receiver.
- [sourceExpression](#) (page 15)
Returns the source expression for the receiver.
- [setSourceExpression:](#) (page 13)
Sets the source expression for the receiver.

Managing Destination Information

- [destinationEntityName](#) (page 7)
Returns the destination entity name for the receiver.
- [setDestinationEntityName:](#) (page 10)
Sets the destination entity name for the receiver.
- [destinationEntityVersionHash](#) (page 8)
Returns the version hash for the destination entity for the receiver.
- [setDestinationEntityVersionHash:](#) (page 10)
Sets the version hash for the destination entity for the receiver.

Managing Mapping Information

- [name](#) (page 9)
Returns the name of the receiver.
- [setName:](#) (page 12)
Sets the name of the receiver.
- [mappingType](#) (page 8)
Returns the mapping type for the receiver.
- [setMappingType:](#) (page 11)
Sets the mapping type for the receiver.
- [entityMigrationPolicyClassName](#) (page 8)
Returns the class name of the migration policy for the receiver.
- [setEntityMigrationPolicyClassName:](#) (page 11)
Sets the class name of the migration policy for the receiver.
- [attributeMappings](#) (page 7)
Returns the array of attribute mappings for the receiver.
- [setAttributeMappings:](#) (page 10)
Sets the array of attribute mappings for the receiver.
- [relationshipMappings](#) (page 9)
Returns the array of relationship mappings for the receiver.
- [setRelationshipMappings:](#) (page 12)
Sets the array of relationship mappings for the receiver.
- [userInfo](#) (page 15)
Returns the user info dictionary for the receiver.
- [setUserInfo:](#) (page 13)
Sets the user info dictionary for the receiver.

Instance Methods

attributeMappings

Returns the array of attribute mappings for the receiver.

- (NSArray *)attributeMappings

Return Value

The array of attribute mappings for the receiver.

Special Considerations

The order of mappings in the array specifies the order in which the mappings will be processed during a migration.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setAttributeMappings:](#) (page 10)
- [relationshipMappings](#) (page 9)

Declared In

NSEntityMapping.h

destinationEntityName

Returns the destination entity name for the receiver.

- (NSString *)destinationEntityName

Return Value

The destination entity name for the receiver.

Discussion

Mappings are not directly bound to entity descriptions. You can use the migration manager's `destinationEntityForEntityMapping:` method to retrieve the entity description for this entity name.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setDestinationEntityName:](#) (page 10)
- [sourceEntityName](#) (page 14)

Declared In

NSEntityMapping.h

destinationEntityVersionHash

Returns the version hash for the destination entity for the receiver.

- (NSData *)destinationEntityVersionHash

Return Value

The version hash for the destination entity for the receiver.

Discussion

The version hash is calculated by Core Data based on the property values of the entity (see `NSEntityDescription`'s `versionHash` method). The `destinationEntityVersionHash` must equal the version hash of the destination entity represented by the mapping.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setDestinationEntityVersionHash:](#) (page 10)
- [sourceEntityVersionHash](#) (page 14)

Declared In

`NSEntityMapping.h`

entityMigrationPolicyClassName

Returns the class name of the migration policy for the receiver.

- (NSString *)entityMigrationPolicyClassName

Return Value

The class name of the migration policy for the receiver.

Discussion

If not specified, the default migration class name is `NSEntityMigrationPolicy`. You can specify a subclass to provide custom behavior.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setEntityMigrationPolicyClassName:](#) (page 11)

Declared In

`NSEntityMapping.h`

mappingType

Returns the mapping type for the receiver.

- (NSEntityMappingType)mappingType

Return Value

The mapping type for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setMappingType:](#) (page 11)

Declared In

NSEntityMapping.h

name

Returns the name of the receiver.

- (NSString *)name

Return Value

The name of the receiver.

Discussion

The name is used only as a means of distinguishing mappings in a model. If not specified, the value defaults to *SOURCE->DESTINATION*.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setName:](#) (page 12)

Declared In

NSEntityMapping.h

relationshipMappings

Returns the array of relationship mappings for the receiver.

- (NSArray *)relationshipMappings

Return Value

The array of relationship mappings for the receiver.

Special Considerations

The order of mappings in the array specifies the order in which the mappings will be processed during a migration.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setRelationshipMappings:](#) (page 12)

- [attributeMappings](#) (page 7)

Declared In

NSEntityMapping.h

setAttributeMappings:

Sets the array of attribute mappings for the receiver.

```
- (void)setAttributeMappings:(NSArray *)mappings
```

Parameters

mappings

The array of attribute mappings for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [attributeMappings](#) (page 7)
- [setRelationshipMappings:](#) (page 12)

Declared In

NSEntityMapping.h

setDestinationEntityName:

Sets the destination entity name for the receiver.

```
- (void)setDestinationEntityName:(NSString *)name
```

Parameters

name

The destination entity name.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [destinationEntityName](#) (page 7)
- [setSourceEntityName:](#) (page 12)

Declared In

NSEntityMapping.h

setDestinationEntityVersionHash:

Sets the version hash for the destination entity for the receiver.

```
- (void)setDestinationEntityVersionHash:(NSData *)vhash
```

Parameters

vhash

The version hash for the destination entity.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [destinationEntityVersionHash](#) (page 8)
- [setSourceEntityVersionHash:](#) (page 13)

Declared In

NSEntityMapping.h

setEntityMigrationPolicyClassName:

Sets the class name of the migration policy for the receiver.

```
- (void)setEntityMigrationPolicyClassName:(NSString *)name
```

Parameters*name*

The class name of the migration policy (either `NSEntityMigrationPolicy` or a subclass of `NSEntityMigrationPolicy`).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [entityMigrationPolicyClassName](#) (page 8)

Declared In

NSEntityMapping.h

setMappingType:

Sets the mapping type for the receiver.

```
- (void)setMappingType:(NSEntityMappingType)type
```

Parameters*type*

The mapping type for the receiver.

Discussion

If you specify a custom entity mapping type, you must specify a value for the migration policy class name as well (see [setEntityMigrationPolicyClassName:](#) (page 11)).

Availability

Available in Mac OS X v10.5 and later.

See Also

- [mappingType](#) (page 8)

Declared In

NSEntityMapping.h

setName:

Sets the name of the receiver.

```
- (void)setName:(NSString *)name
```

Parameters

name

The name of the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [name](#) (page 9)

Declared In

NSEntityMapping.h

setRelationshipMappings:

Sets the array of relationship mappings for the receiver.

```
- (void)setRelationshipMappings:(NSArray *)mappings
```

Parameters

mappings

The array of relationship mappings for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [relationshipMappings](#) (page 9)
- [setAttributeMappings:](#) (page 10)

Declared In

NSEntityMapping.h

setSourceEntityName:

Sets the source entity name for the receiver.

```
- (void)setSourceEntityName:(NSString *)name
```

Parameters

name

The source entity name for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [sourceEntityName](#) (page 14)

- [setDestinationEntityName:](#) (page 10)

Declared In

NSEntityMapping.h

setSourceEntityVersionHash:

Sets the version hash for the source entity for the receiver.

- (void)setSourceEntityVersionHash:(NSData *)*vhash*

Parameters

vhash

The version hash for the source entity for the receiver.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [sourceEntityVersionHash](#) (page 14)

- [setDestinationEntityVersionHash:](#) (page 10)

Declared In

NSEntityMapping.h

setSourceExpression:

Sets the source expression for the receiver.

- (void)setSourceExpression:(NSEExpression *)*source*

Parameters

source

The source expression for the receiver. The expression can be a fetch request expression, or any other expression which evaluates to a collection.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [sourceExpression](#) (page 15)

Declared In

NSEntityMapping.h

setUserInfo:

Sets the user info dictionary for the receiver.

- (void)setUserInfo:(NSDictionary *)*dict*

Parameters*dict*

The user info dictionary for the receiver.

Discussion

You can set the contents of the dictionary using the appropriate inspector in the Xcode mapping model editor.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [userInfo](#) (page 15)

Declared In

NSEntityMapping.h

sourceEntityName

Returns the source entity name for the receiver.

```
- (NSString *)sourceEntityName
```

Return Value

The source entity name for the receiver.

Discussion

Mappings are not directly bound to entity descriptions; you can use the `sourceEntityForEntityMapping:` method on the migration manager to retrieve the entity description for this entity name.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setSourceEntityName:](#) (page 12)

- [destinationEntityName](#) (page 7)

Declared In

NSEntityMapping.h

sourceEntityVersionHash

Returns the version hash for the source entity for the receiver.

```
- (NSData *)sourceEntityVersionHash
```

Return Value

The version hash for the source entity for the receiver.

Discussion

The version hash is calculated by Core Data based on the property values of the entity (see `NSEntityDescription's versionHash` method). The `sourceEntityVersionHash` must equal the version hash of the source entity represented by the mapping.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setSourceEntityVersionHash:](#) (page 13)
- [destinationEntityVersionHash](#) (page 8)

Declared In

NSEntityMapping.h

sourceExpression

Returns the source expression for the receiver.

- (NSExpression *)sourceExpression

Return Value

The source expression. The expression can be a fetch request expression, or any other expression which evaluates to a collection.

Discussion

The source expression is used to obtain the collection of managed objects to process through the mapping.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setSourceExpression:](#) (page 13)

Declared In

NSEntityMapping.h

userInfo

Returns the user info dictionary for the receiver.

- (NSDictionary *)userInfo

Return Value

The user info dictionary.

Discussion

You can use the info dictionary in any way that might be useful in your migration. You set the contents of the dictionary using [setUserInfo:](#) (page 13) or using the appropriate inspector in the Xcode mapping model editor.

Availability

Available in Mac OS X v10.5 and later.

See Also

- [setUserInfo:](#) (page 13)

Declared In

NSEntityMapping.h

Constants

Entity Mapping Types

These constants specify the types of entity mapping.

```
enum {
    NSUndefinedEntityType      = 0x00,
    NSCustomEntityType         = 0x01,
    NSAddEntityType            = 0x02,
    NSRemoveEntityType         = 0x03,
    NSCopyEntityType           = 0x04,
    NSTransformEntityType       = 0x05
};
```

Constants

NSUndefinedEntityType

Specifies that the developer handles destination instance creation.

Available in Mac OS X v10.5 and later.

Declared in NSEntityMapping.h.

NSCustomEntityType

Specifies a custom mapping.

Available in Mac OS X v10.5 and later.

Declared in NSEntityMapping.h.

NSAddEntityType

Specifies that this is a new entity in the destination model.

Instances of the entity only exist in the destination.

Available in Mac OS X v10.5 and later.

Declared in NSEntityMapping.h.

NSRemoveEntityType

Specifies that this entity is not present in the destination model.

Instances of the entity only exist in the source—source instances are not mapped to destination.

Available in Mac OS X v10.5 and later.

Declared in NSEntityMapping.h.

NSCopyEntityType

Specifies that source instances are migrated as-is.

Available in Mac OS X v10.5 and later.

Declared in NSEntityMapping.h.

NSTransformEntityType

Specifies that entity exists in source and destination and is mapped.

Available in Mac OS X v10.5 and later.

Declared in NSEntityMapping.h.

Declared In

NSEntityMapping.h

NSEntityMappingType

Data type used for constants that specify types of entity mapping.

```
typedef NSUInteger NSEntityMappingType;
```

Discussion

For possible values, see [“Entity Mapping Types”](#) (page 16).

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSEntityMapping.h

Document Revision History

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

Date	Notes
2007-05-22	New document that describes the class used to specify the mapping of an entity from one version of a Core Data schema to another.

REVISION HISTORY

Document Revision History

Index

A

attributeMappings [instance method 7](#)

D

destinationEntityName [instance method 7](#)
destinationEntityVersionHash [instance method 8](#)

E

Entity Mapping Types [16](#)
entityMigrationPolicyClassName [instance method 8](#)

M

mappingType [instance method 8](#)

N

name [instance method 9](#)
NSAddEntityType [constant 16](#)
NSCopyEntityType [constant 16](#)
NSCustomEntityType [constant 16](#)
NSEntityMappingType [data type 17](#)
NSRemoveEntityType [constant 16](#)
NSTransformEntityType [constant 16](#)
NSUndefinedEntityType [constant 16](#)

R

relationshipMappings [instance method 9](#)

S

setAttributeMappings: [instance method 10](#)
setDestinationEntityName: [instance method 10](#)
setDestinationEntityVersionHash: [instance method 10](#)
setEntityMigrationPolicyClassName: [instance method 11](#)
setMappingType: [instance method 11](#)
setName: [instance method 12](#)
setRelationshipMappings: [instance method 12](#)
setSourceEntityName: [instance method 12](#)
setSourceEntityVersionHash: [instance method 13](#)
setSourceExpression: [instance method 13](#)
setUserInfo: [instance method 13](#)
sourceEntityName [instance method 14](#)
sourceEntityVersionHash [instance method 14](#)
sourceExpression [instance method 15](#)

U

userInfo [instance method 15](#)