

---

# NSMigrationManager Class Reference

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



2007-05-22



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, 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

---

## **NSMigrationManager Class Reference 5**

---

Overview 5

Tasks 5

    Initializing a Manager 5

    Performing Migration Operations 5

    Monitoring Migration Progress 6

    Working with Source and Destination Instances 6

    Getting Information About a Migration Manager 6

Instance Methods 6

    associateSourceInstance:withDestinationInstance:forEntityMapping: 6

    cancelMigrationWithError: 7

    currentEntityMapping 8

    destinationContext 8

    destinationEntityForEntityMapping: 8

    destinationInstancesForEntityMappingNamed:sourceInstances: 9

    destinationModel 9

    initWithSourceModel:destinationModel: 10

    mappingModel 11

    migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:

    destinationType:destinationOptions:error: 11

    migrationProgress 12

    reset 12

    setUserInfo: 13

    sourceContext 13

    sourceEntityForEntityMapping: 13

    sourceInstancesForEntityMappingNamed:destinationInstances: 14

    sourceModel 15

    userInfo 15

---

## **Document Revision History 17**

---

## **Index 19**

---



# NSMigrationManager Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/CoreData.framework
<b>Availability</b>	Available in Mac OS X v10.5 and later.
<b>Companion guide</b>	Core Data Model Versioning and Data Migration Programming Guide
<b>Declared in</b>	NSMigrationManager.h

## Overview

Instances of `NSMigrationManager` perform a migration of data from one persistent store to another using a given mapping model.

## Tasks

### Initializing a Manager

- [initWithSourceModel:destinationModel:](#) (page 10)  
Initializes a migration manager instance with given source and destination models.
- [setUserInfo:](#) (page 13)  
Sets the user info for the receiver.

### Performing Migration Operations

- [migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:](#) (page 11)  
Migrates the store at a given source URL to the store at a given destination URL, performing all of the mappings specified in a given mapping model.
- [reset](#) (page 12)  
Resets the association tables for the migration.
- [cancelMigrationWithError:](#) (page 7)  
Cancels the migration with a given error.

## Monitoring Migration Progress

- [migrationProgress](#) (page 12)  
Returns a number from 0 to 1 that indicates the proportion of completeness of the migration.
- [currentEntityMapping](#) (page 8)  
Returns the entity mapping currently being processed.

## Working with Source and Destination Instances

- [associateSourceInstance:withDestinationInstance:forEntityMapping:](#) (page 6)  
Associates a given source instance with an array of destination instances for a given property mapping.
- [destinationInstancesForEntityMappingNamed:sourceInstances:](#) (page 9)  
Returns the managed object instances created in the destination store for a named entity mapping for a given array of source instances.
- [sourceInstancesForEntityMappingNamed:destinationInstances:](#) (page 14)  
Returns the managed object instances in the source store used to create a given destination instance for a given property mapping.

## Getting Information About a Migration Manager

- [mappingModel](#) (page 11)  
Returns the mapping model for the receiver.
- [sourceModel](#) (page 15)  
Returns the source model for the receiver.
- [destinationModel](#) (page 9)  
Returns the destination model for the receiver.
- [sourceEntityForEntityMapping:](#) (page 13)  
Returns the entity description for the source entity of a given entity mapping.
- [destinationEntityForEntityMapping:](#) (page 8)  
Returns the entity description for the destination entity of a given entity mapping.
- [sourceContext](#) (page 13)  
Returns the managed object context the receiver uses for reading the source persistent store.
- [destinationContext](#) (page 8)  
Returns the managed object context the receiver uses for writing the destination persistent store.
- [userInfo](#) (page 15)  
Returns the user info for the receiver.

## Instance Methods

### **associateSourceInstance:withDestinationInstance:forEntityMapping:**

Associates a given source instance with an array of destination instances for a given property mapping.

```
- (void)associateSourceInstance:(NSManagedObject *)sourceInstance
    withDestinationInstance:(NSManagedObject *)destinationInstance
    forEntityMapping:(NSEntityMapping *)entityMapping
```

**Parameters**

*sourceInstance*

A source managed object.

*destinationInstance*

The destination manage object for *sourceInstance*.

*entityMapping*

The entity mapping to use to associate *sourceInstance* with the object in *destinationInstances*.

**Discussion**

Data migration is performed as a three-stage process (first create the data, then relate the data, then validate the data). You use this method to associate data between the source and destination stores, in order to allow for relationship creation or fix-up after the creation stage.

This method is called in the default implementation of `NSEntityMigrationPolicy's createDestinationInstancesForSourceInstance:entityMapping:manager:error:` method.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [sourceInstancesForEntityMappingNamed:destinationInstances:](#) (page 14)
- [destinationInstancesForEntityMappingNamed:sourceInstances:](#) (page 9)

**Declared In**

`NSMigrationManager.h`

**cancelMigrationWithError:**

Cancels the migration with a given error.

```
- (void)cancelMigrationWithError:(NSError *)error
```

**Parameters**

*error*

An error object that describes the reason why the migration is canceled.

**Discussion**

You can invoke this method from anywhere in the migration process to abort the migration. Calling this method causes `migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:` (page 11) to abort the migration and return *error*—you should provide an appropriate error to indicate the reason for the cancellation.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`NSMigrationManager.h`

## currentEntityMapping

Returns the entity mapping currently being processed.

- (NSEntityMapping \*)currentEntityMapping

### Return Value

The entity mapping currently being processed.

### Discussion

Each entity is processed a total of three times (instance creation, relationship creation, validation).

### Special Considerations

You can observe this value using key-value observing.

### Availability

Available in Mac OS X v10.5 and later.

### Declared In

NSMigrationManager.h

## destinationContext

Returns the managed object context the receiver uses for writing the destination persistent store.

- (NSManagedObjectContext \*)destinationContext

### Return Value

The managed object context the receiver uses for writing the destination persistent store.

### Discussion

This context is created lazily as part of the initialization of the Core Data stacks used for migration.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [sourceContext](#) (page 13)

### Declared In

NSMigrationManager.h

## destinationEntityForEntityMapping:

Returns the entity description for the destination entity of a given entity mapping.

- (NSEntityDescription \*)destinationEntityForEntityMapping:(NSEntityMapping \*)*mEntity*

### Parameters

*mEntity*

An entity mapping.

### Return Value

The entity description for the destination entity of *mEntity*.



**Discussion**

Entity mappings do not store the actual description objects, but rather the name and version information of the entity.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

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

**Declared In**

NSMigrationManager.h

**destinationInstancesForEntityMappingNamed:sourceInstances:**

Returns the managed object instances created in the destination store for a named entity mapping for a given array of source instances.

```
- (NSArray *)destinationInstancesForEntityMappingNamed:(NSString *)mappingName
    sourceInstances:(NSArray *)sourceInstances
```

**Parameters**

*mappingName*

The name of an entity mapping in use.

*sourceInstances*

A array of managed objects in the source store.

**Return Value**

An array containing the managed object instances created in the destination store for the entity mapping named *mappingName* for *sourceInstances*. If *sourceInstances* is `nil`, all of the destination instances created by the specified property mapping are returned.

**Special Considerations**

This method throws an `NSInvalidArgumentException` exception if *mappingName* is not a valid mapping name.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [sourceInstancesForEntityMappingNamed:destinationInstances:](#) (page 14)

**Declared In**

NSMigrationManager.h

**destinationModel**

Returns the destination model for the receiver.

```
- (NSManagedObjectModel *)destinationModel
```

**Return Value**

The destination model for the receiver.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [mappingModel](#) (page 11)
- [sourceModel](#) (page 15)
- [initWithSourceModel:destinationModel:](#) (page 10)

**Declared In**

NSMigrationManager.h

**initWithSourceModel:destinationModel:**

Initializes a migration manager instance with given source and destination models.

```
- (id)initWithSourceModel:(NSManagedObjectModel *)sourceModel
    destinationModel:(NSManagedObjectModel *)destinationModel
```

**Parameters**

*sourceModel*

The source managed object model for the migration manager.

*destinationModel*

The destination managed object model for the migration manager.

**Return Value**

A migration manager instance initialized to migrate data in a store that uses *sourceModel* to a store that uses *destinationModel*.

**Discussion**

You specify the mapping model in the migration method,

[migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:](#) (page 11).

**Special Considerations**

This is the designated initializer for NSMigrationManager.

Although validation of the models is performed during

[migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:](#) (page 11), as with

NSPersistentStoreCoordinator once models are added to the migration manager they are immutable and cannot be altered.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:](#) (page 11)
- [mappingModel](#) (page 11)
- [sourceModel](#) (page 15)
- [destinationModel](#) (page 9)

**Declared In**

NSMigrationManager.h

## mappingModel

Returns the mapping model for the receiver.

- (NSMappingModel \*)mappingModel

### Return Value

The mapping model for the receiver.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [sourceModel](#) (page 15)

- [destinationModel](#) (page 9)

- [migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:](#) (page 11)

### Declared In

NSMigrationManager.h

## migrateStoreFromURL:type:options:withMappingModel:toDestinationURL:destinationType:destinationOptions:error:

Migrates of the store at a given source URL to the store at a given destination URL, performing all of the mappings specified in a given mapping model.

```
- (BOOL)migrateStoreFromURL:(NSURL *)sourceURL
    type:(NSString *)sStoreType
    options:(NSDictionary *)sOptions
    withMappingModel:(NSMappingModel *)mappings
    toDestinationURL:(NSURL *)dURL
    destinationType:(NSString *)dStoreType
    destinationOptions:(NSDictionary *)dOptions
    error:(NSError **)error
```

### Parameters

*sourceURL*

The location of an existing persistent store. A store must exist at this URL.

*sStoreType*

The type of store at *sourceURL* (see [NSPersistentStoreCoordinator](#) for possible values).

*sOptions*

A dictionary of options for the source (see [NSPersistentStoreCoordinator](#) for possible values).

*mappings*

The mapping model to use to effect the migration.

*dURL*

The location of the destination store.

*dStoreType*

The type of store at *dURL* (see [NSPersistentStoreCoordinator](#) for possible values).

*dOptions*

A dictionary of options for the destination (see `NSPersistentStoreCoordinator` for possible values).

*error*

If an error occurs during the validation or migration, upon return contains an `NSError` object that describes the problem.

**Return Value**

YES if the migration proceeds without errors during the compatibility checks or migration, otherwise NO.

**Discussion**

This method performs compatibility checks on the source and destination models and the mapping model.

**Special Considerations**

If a store does not exist at the destination URL (*dURL*), one is created; otherwise, the migration appends to the existing store.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [cancelMigrationWithError:](#) (page 7)

**Declared In**

`NSMigrationManager.h`

**migrationProgress**

Returns a number from 0 to 1 that indicates the proportion of completeness of the migration.

- (float)migrationProgress

**Return Value**

A number from 0 to 1 that indicates the proportion of completeness of the migration. If a migration is not taking place, returns 1.

**Special Considerations**

You can observe this value using key-value observing.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`NSMigrationManager.h`

**reset**

Resets the association tables for the migration.

- (void)reset

**Special Considerations**

This method does not reset the source or destination contexts.

### Availability

Available in Mac OS X v10.5 and later.

### Declared In

NSMigrationManager.h

## setUserInfo:

Sets the user info for the receiver.

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

### Parameters

*dict*

The user info for the receiver.

### Discussion

You can use the user info dictionary to aid the customization of your migration process.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [userInfo](#) (page 15)

### Declared In

NSMigrationManager.h

## sourceContext

Returns the managed object context the receiver uses for reading the source persistent store.

```
- (NSManagedObjectContext *)sourceContext
```

### Return Value

The managed object context the receiver uses for reading the source persistent store.

### Discussion

This context is created lazily as part of the initialization of the Core Data stacks used for migration.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [destinationContext](#) (page 8)

### Declared In

NSMigrationManager.h

## sourceEntityForEntityMapping:

Returns the entity description for the source entity of a given entity mapping.

- (NSEntityDescription \*)sourceEntityForEntityMapping:(NSEntityMapping \*)*mEntity*

#### Parameters

*mEntity*

An entity mapping.

#### Return Value

The entity description for the source entity of *mEntity*.

#### Discussion

Entity mappings do not store the actual description objects, but rather the name and version information of the entity.

#### Availability

Available in Mac OS X v10.5 and later.

#### See Also

- [destinationEntityForEntityMapping:](#) (page 8)

#### Declared In

NSMigrationManager.h

### sourceInstancesForEntityMappingNamed:destinationInstances:

Returns the managed object instances in the source store used to create a given destination instance for a given property mapping.

- (NSArray \*)sourceInstancesForEntityMappingNamed:(NSString \*)*mappingName*  
destinationInstances:(NSArray \*)*destinationInstances*

#### Parameters

*mappingName*

The name of an entity mapping in use.

*destinationInstances*

A array of managed objects in the destination store.

#### Return Value

An array containing the managed object instances in the source store used to create *destinationInstances* using the entity mapping named *mappingName*. If *destinationInstances* is nil, all of the source instances used to create the destination instance for this property mapping are returned.

#### Special Considerations

This method throws an `NSInvalidArgumentException` exception if *mappingName* is not a valid mapping name.

#### Availability

Available in Mac OS X v10.5 and later.

#### See Also

- [destinationInstancesForEntityMappingNamed:sourceInstances:](#) (page 9)

#### Declared In

NSMigrationManager.h

## sourceModel

Returns the source model for the receiver.

- (NSManagedObjectModel \*)sourceModel

### Return Value

The source model for the receiver.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

- [mappingModel](#) (page 11)
- [destinationModel](#) (page 9)
- [initWithSourceModel:destinationModel:](#) (page 10)

### Declared In

NSMigrationManager.h

## userInfo

Returns the user info for the receiver.

- (NSDictionary \*)userInfo

### Return Value

The user info for the receiver.

### Availability

Available in Mac OS X v10.5 and later.

### See Also

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

### Declared In

NSMigrationManager.h





# Document Revision History

---

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

Date	Notes
2007-05-22	New document that describes the Core Data class that performs a migration of data from one persistent store to another.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

associateSourceInstance:withDestinationInstance:  
forEntityMapping: [instance method 6](#)

## C

---

cancelMigrationWithError: [instance method 7](#)  
currentEntityMapping [instance method 8](#)

## D

---

destinationContext [instance method 8](#)  
destinationEntityForEntityMapping: [instance method 8](#)  
destinationInstancesForEntityMappingNamed:  
sourceInstances: [instance method 9](#)  
destinationModel [instance method 9](#)

## I

---

initWithSourceModel:destinationModel: [instance method 10](#)

## M

---

mappingModel [instance method 11](#)  
migrateStoreFromURL:type:options:withMappingModel:  
toDestinationURL:destinationType:  
destinationOptions:error: [instance method 11](#)  
migrationProgress [instance method 12](#)

## R

---

reset [instance method 12](#)

## S

---

setUserInfo: [instance method 13](#)  
sourceContext [instance method 13](#)  
sourceEntityForEntityMapping: [instance method 13](#)  
sourceInstancesForEntityMappingNamed:  
destinationInstances: [instance method 14](#)  
sourceModel [instance method 15](#)

## U

---

userInfo [instance method 15](#)