Xcode Mapping Tool for Core Data

Tools > Xcode

2007-12-30

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

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

Figures

Introduction

The purpose of the mapping model tool is to create a mapping model for use with the Core Data framework to support data migration. For a discussion of data migration, see *Core Data Model Versioning and Data Migration Programming Guide*; for more about Core Data, see *Core Data Programming Guide*.

You should read this document to understand the features of the Xcode mapping model tool, the fundamentals of how to create a mapping model and how to use the tool to edit the model.

Important: This chapter only describes the function of the mapping model tool, it does not explain the features of data migration or how to use the model itself. You must read *Core Data Model Versioning and Data Migration Programming Guide* before reading this document or attempting to use the mapping model tool.

Organization of This Document

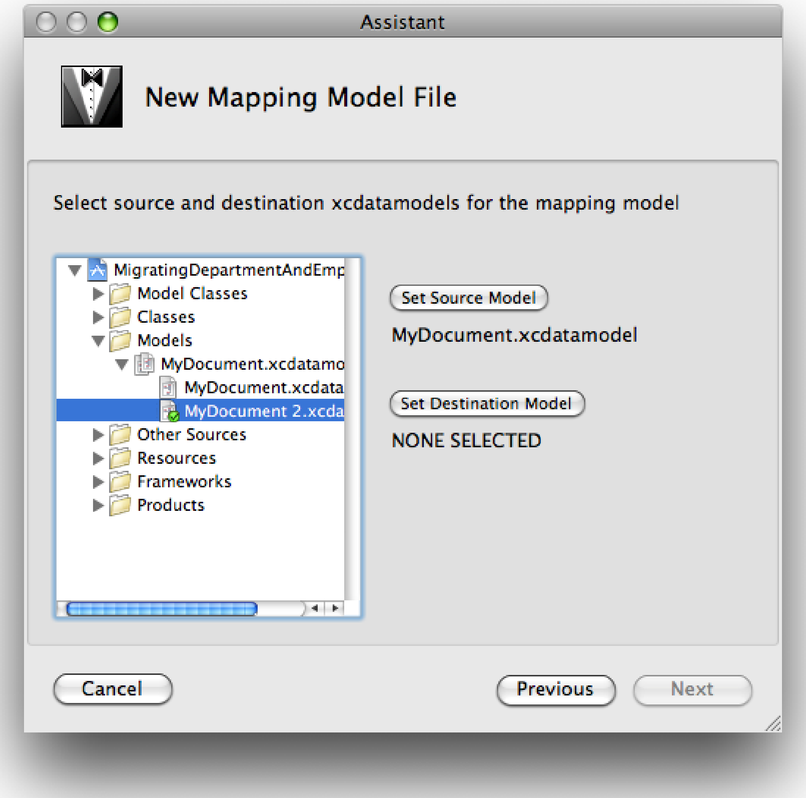
The document is organized into the following sections:

- "Creating and Updating a Mapping Model" (page 9) describes how to create and update a mapping model.
- "The Mapping Model Pane" (page 11) describes the features of the mapping model tool.
- "Displaying Differences Between Models" (page 17) describes how you can show the differences between two mapping models.

Creating and Updating a Mapping Model

A mapping model is an instance of `NSMappingModel` with a collection of `NSEntityMapping` and `NSPropertyMapping` objects, together with supporting objects such as instances of `NSEntityMigrationPolicy` and `NSFetchRequest`. (For more about specific Core Data classes, see the relevant API reference documentation.) You can create a model directly in code at runtime; however, it is typically easier to do so graphically using the Mapping tool. (This is analogous to Interface Builder. With Interface Builder, you graphically create a collection of objects that are then saved in a file and re-created at runtime. Similarly, just as you can modify the user interface after it has been loaded, you can customize a mapping model after it has been loaded.)

To create a new mapping model, in Xcode you choose File > New File, then in the New File panel you select Design > Mapping Model and press Next. In the next pane, you give the model a name, and press Next to display the source and destination model selection pane:



Select source and destination xcdatamodels for the mapping model

MyDocument.xcdatamodel

NONE SELECTED

The outline view displays the groups and files in your project. You navigate to an existing model file, then press Set Source Model or Set Destination Model as appropriate. When you have selected both a source and a destination model, you press Finish to create a default mapping model.

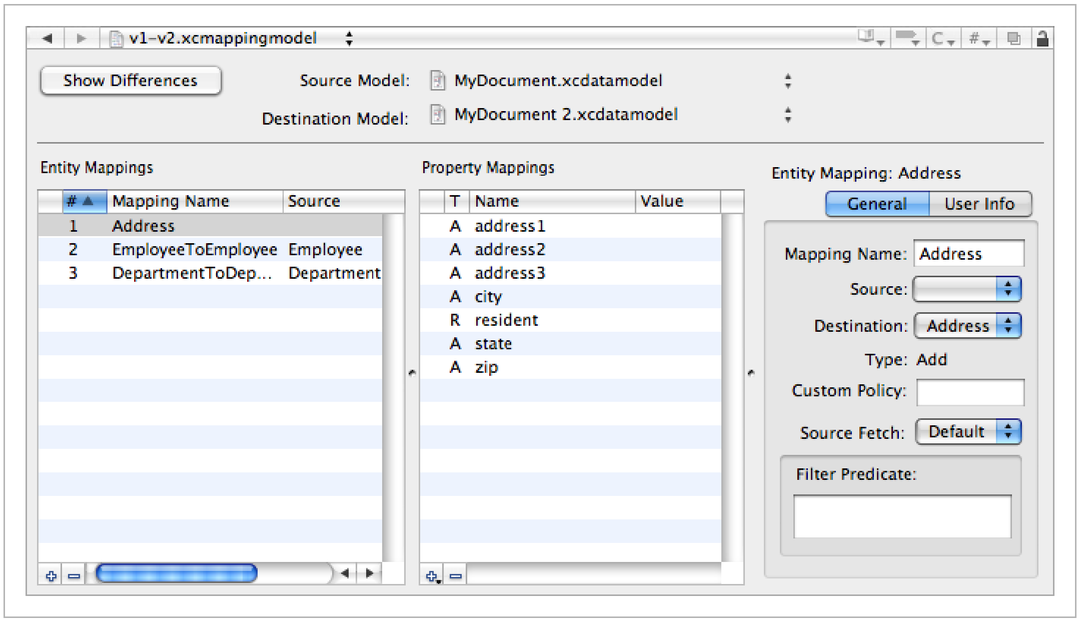
If the source or destination model changes, you can update the mapping model by choosing Design > Mapping Model > Update Source Model or Update Destination Model respectively. (Typically, however, the source and destination models should be finalized before you create the mapping model.)

The Mapping Model Pane

This section describes the mapping model tool pane.

Basics

The mapping model pane consists of two main sections: the upper area that shows the names of the source and destination model files, and the lower area that displays information about the mappings.



Source Model: MyDocument.xcdatamodel

Destination Model: MyDocument 2.xcdatamodel

#	Mapping Name	Source
1	Address	
2	EmployeeToEmployee	Employee
3	DepartmentToDep…	Department

T	Name	Value
A	address1	
A	address2	
A	address3	
A	city	
R	resident	
A	state	
A	zip	

Entity Mapping: Address

Mapping Name: Address

Source:

Destination: Address

Type: Add

Custom Policy:

Source Fetch: Default

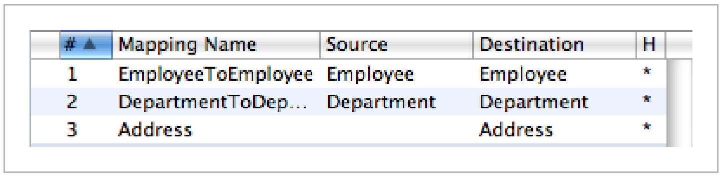
You press the Show Differences button to display a panel that highlights the differences between the source and destination models. This is described further in "Displaying Differences Between Models" (page 17).

You use the source and destination popup menus to show the paths to the model files, or to select new source or destination model files.

You use the mappings panes to display and edit information about the mappings in the mapping model.

The Entity Mappings Pane

Figure 1 The entity mappings pane



# ▲	Mapping Name	Source	Destination	H	
1	EmployeeToEmployee	Employee	Employee	*	
2	DepartmentToDep...	Department	Department	*	
3	Address		Address	*	

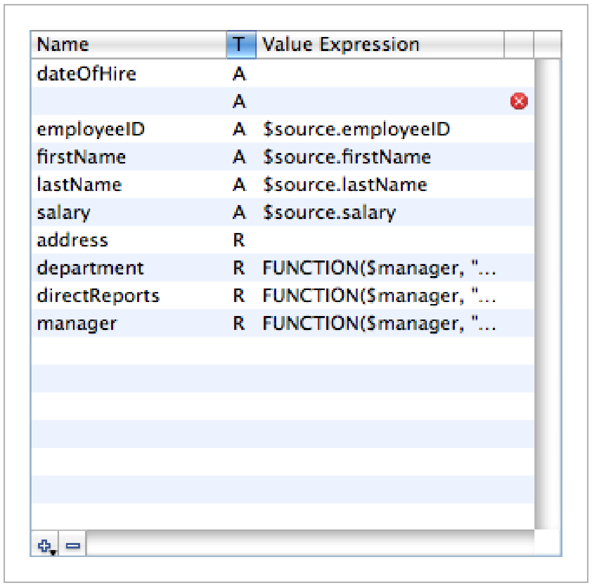
The entity mappings pane lists the entity mappings in the model—each row corresponds to an instance of `NSEntityMapping` in the mapping model. The columns show the index, name, source entity, destination entity, and hash difference of each mapping. The "hash difference" ("H") column indicates whether there are differences in the hash values of the source and destination (a "*" indicates that there are, otherwise the cell is empty). The blank column is the error indicator column. If there is a problem with the mapping, the column displays an "X" in a red circle (shown in Figure 2).

You add and delete entity mappings using the plus (+) and minus (–) buttons at the left of the horizontal scroller. You cannot edit the mapping attributes directly in the mappings pane—instead you use the detail pane. The order of the entity mappings (as denoted by their indexes), however, may be important (see *Core Data Model Versioning and Data Migration Programming Guide*). You can rearrange the mappings by dragging the rows.

The Property Mappings Pane

The property mappings pane shows the property mappings for the selected entity mapping. Each row corresponds to an instance of `NSPropertyMapping` in the model.

Figure 2 The property mappings pane



Name	T	Value Expression	
dateOfHire	A		
	A		⊗
employeeID	A	$source.employeeID	
firstName	A	$source.firstName	
lastName	A	$source.lastName	
salary	A	$source.salary	
address	R		
department	R	FUNCTION($manager, "...	
directReports	R	FUNCTION($manager, "...	
manager	R	FUNCTION($manager, "...	

There are four columns showing: the name of the property in the destination entity, the property type ("A" denotes an attribute, "R" denotes a relationship), and the value expression (if any) for the mapping. The blank column is the error indicator column. If there is a problem with the mapping, the column displays an "X" in a red circle.

You add and delete property mappings using the plus (+) and minus (–) buttons respectively at the left of the horizontal scroller. When you add a mapping, you specify whether it is for an attribute or for a relationship. You cannot edit the mapping attributes directly in the mappings pane—instead you use the detail pane.

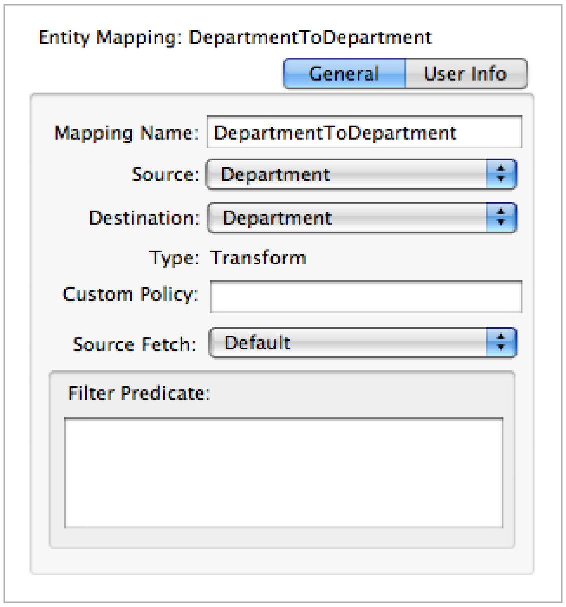
The Detail Pane

The detail pane shows the details for the most-recently selected entity or property mapping. There are two views in the detail pane, the general view and user info view. There are two forms of the general view, for an entity and for a property mapping; the user info view is the same for both mapping types.

Entity Mapping Detail

The entity mapping detail view shows the name of the mapping, the source and destination entities (if any), the type of mapping (add, delete, transform, or copy), the name of the custom policy class for the mapping (the name of a subclass of `NSEntityMigrationPolicy`), and details of the source expression.

Figure 3 The Entity mapping detail



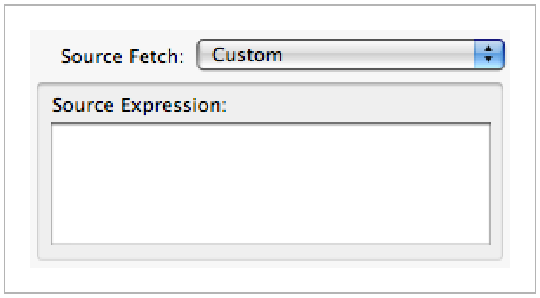
Entity Mapping: DepartmentToDepartment

Mapping Name: DepartmentToDepartment
Source: Department
Destination: Department
Type: Transform
Custom Policy:
Source Fetch: Default
Filter Predicate:

You can edit most of the details directly by typing in the text fields or making an appropriate selection using the popup menu. The exception is the type description, which is a read-only text field that is updated automatically based on the selection of source and destination entities.

You specify the `sourceExpression` for the mapping using a combination of the the source fetch popup menu and the text view beneath it. You can select either the default fetch—in which case you can optionally specify a filter predicate for the fetch—or a custom fetch, in which case you specify the source expression directly.

Figure 4 Source expression detail



Source Fetch: Custom

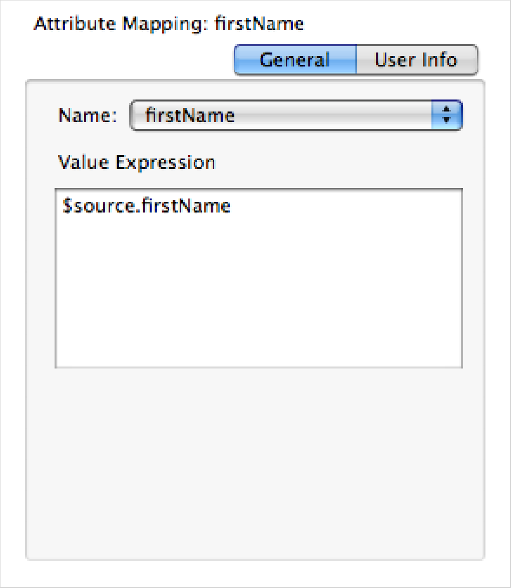
Source Expression:

Property Mapping Detail

The property mapping detail view is different for attributes and properties.

The attribute mapping view shows the name of the attribute in the destination entity in a popup menu and the value expression (if any) associated with the mapping.

Figure 5 The attribute mapping detail



Attribute Mapping: firstName

Name: firstName

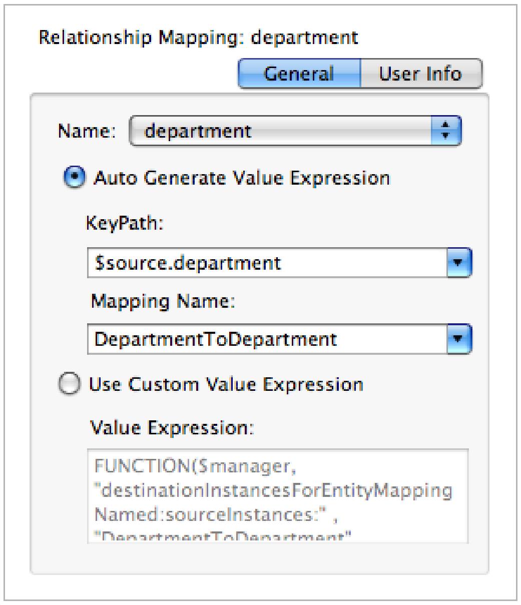
Value Expression

$source.firstName

To edit an attribute mapping, you select an attribute name from the popup menu and type directly in to the value expression text view.

The property mapping view shows the name of the relationship in the destination entity and the value expression for the mapping.

Figure 6 The relationship mapping detail



Relationship Mapping: department

Name: department

● Auto Generate Value Expression

KeyPath: $source.department

Mapping Name: DepartmentToDepartment

○ Use Custom Value Expression

Value Expression:

```
FUNCTION($manager,
"destinationInstancesForEntityMapping
Named:sourceInstances:" ,
"DepartmentToDepartment"
```

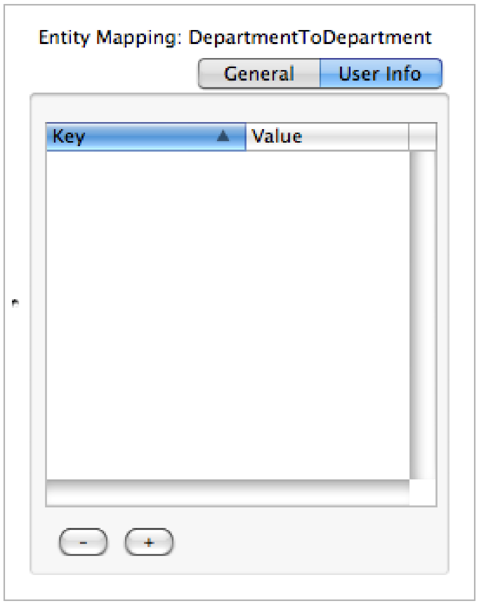
Using the radio buttons, you can select either an auto-generated expression based on the key path of the source relationship and the name of the entity mapping to use, or you can type a custom value expression directly into the value expression text field.

Reserved words: To escape reserved words such as `SIZE`, `FIRST`, and `LAST`, you need to escape them using a `#`, for example `$source.#size`.

User Info

The user info view is the same for entity and property mappings. You can add any key-value pairs you want to the mapping by pressing the plus (+) button and editing the values directly in the table view. You remove key-value pairs by pressing the minus (–) key.

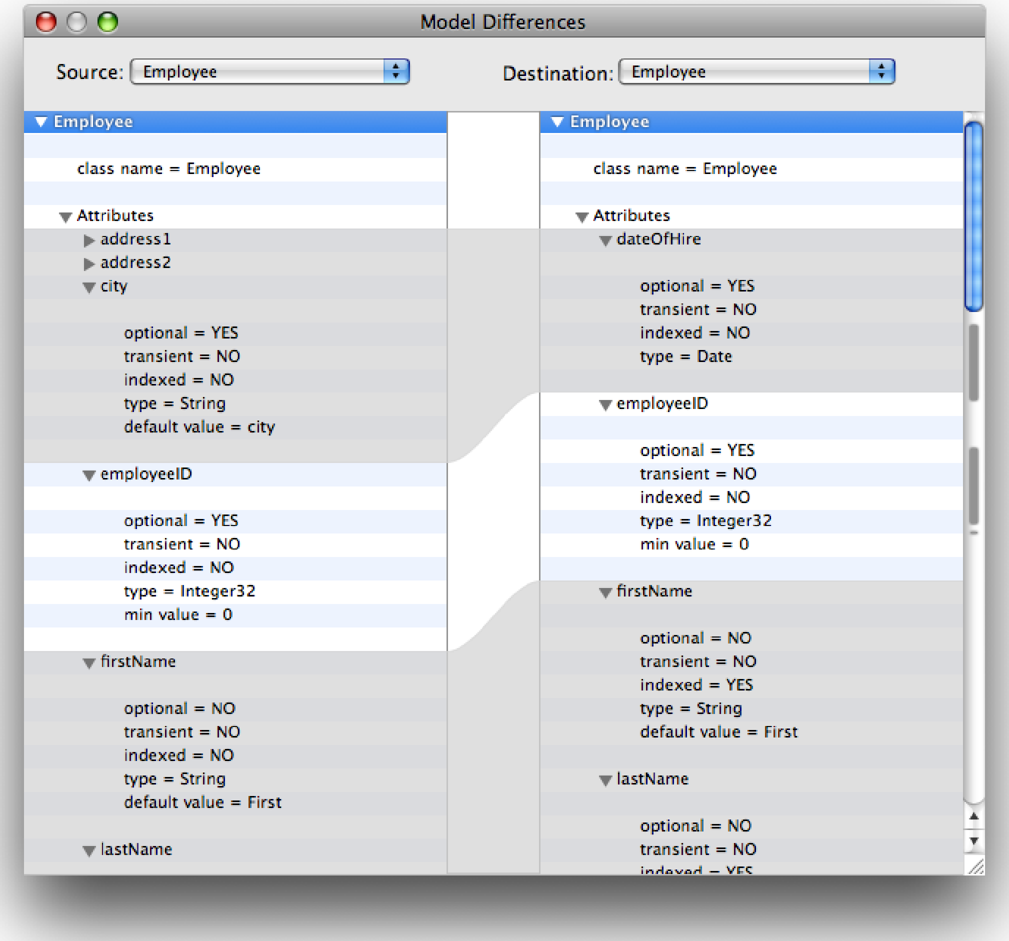
Figure 7 The user info view



Entity Mapping: DepartmentToDepartment

Displaying Differences Between Models

In the main pane, you can press Show Differences to display the differences between the source and destination models. The popup menus allow you to select for the source and destination either all the entities or a single entity to display. In the main pane, the gray areas denote entities and properties that differ between source and destination. You can collapse and expand elements of the display by clicking the associated disclosure triangle.



Source: Employee

Destination: Employee

```
▼ Employee

    class name = Employee

  ▼ Attributes
    ▶ address1
    ▶ address2
    ▼ city

        optional = YES
        transient = NO
        indexed = NO
        type = String
        default value = city

    ▼ employeeID

        optional = YES
        transient = NO
        indexed = NO
        type = Integer32
        min value = 0

    ▼ firstName

        optional = NO
        transient = NO
        indexed = NO
        type = String
        default value = First

    ▼ lastName
```

```
▼ Employee

    class name = Employee

  ▼ Attributes
    ▼ dateOfHire

        optional = YES
        transient = NO
        indexed = NO
        type = Date

    ▼ employeeID

        optional = YES
        transient = NO
        indexed = NO
        type = Integer32
        min value = 0

    ▼ firstName

        optional = NO
        transient = NO
        indexed = YES
        type = String
        default value = First

    ▼ lastName

        optional = NO
        transient = NO
        indexed = YES
```

Document Revision History

This table describes the changes to *Xcode Mapping Tool for Core Data*.

Date	Notes
2007-12-30	New document that describes the Xcode mapping tool for Core Data.