Core Data

Justin Miller • PDX CocoaHeads

What is Core Data?

- Data modeler (*plan & build*)
- Object graph manager (*manage*)
- Persistence framework (*store*)
- Data interface (*query & observe*)

Where can I use it?

- Mac OS X: 10.4 & later
- iOS: 3.0 & later
- Doesn’t necessarily need a UI component

What it is not

- RDBMS (MySQL, PostgreSQL, SQLite, etc.)
- SQLite frontend
- Code eliminator

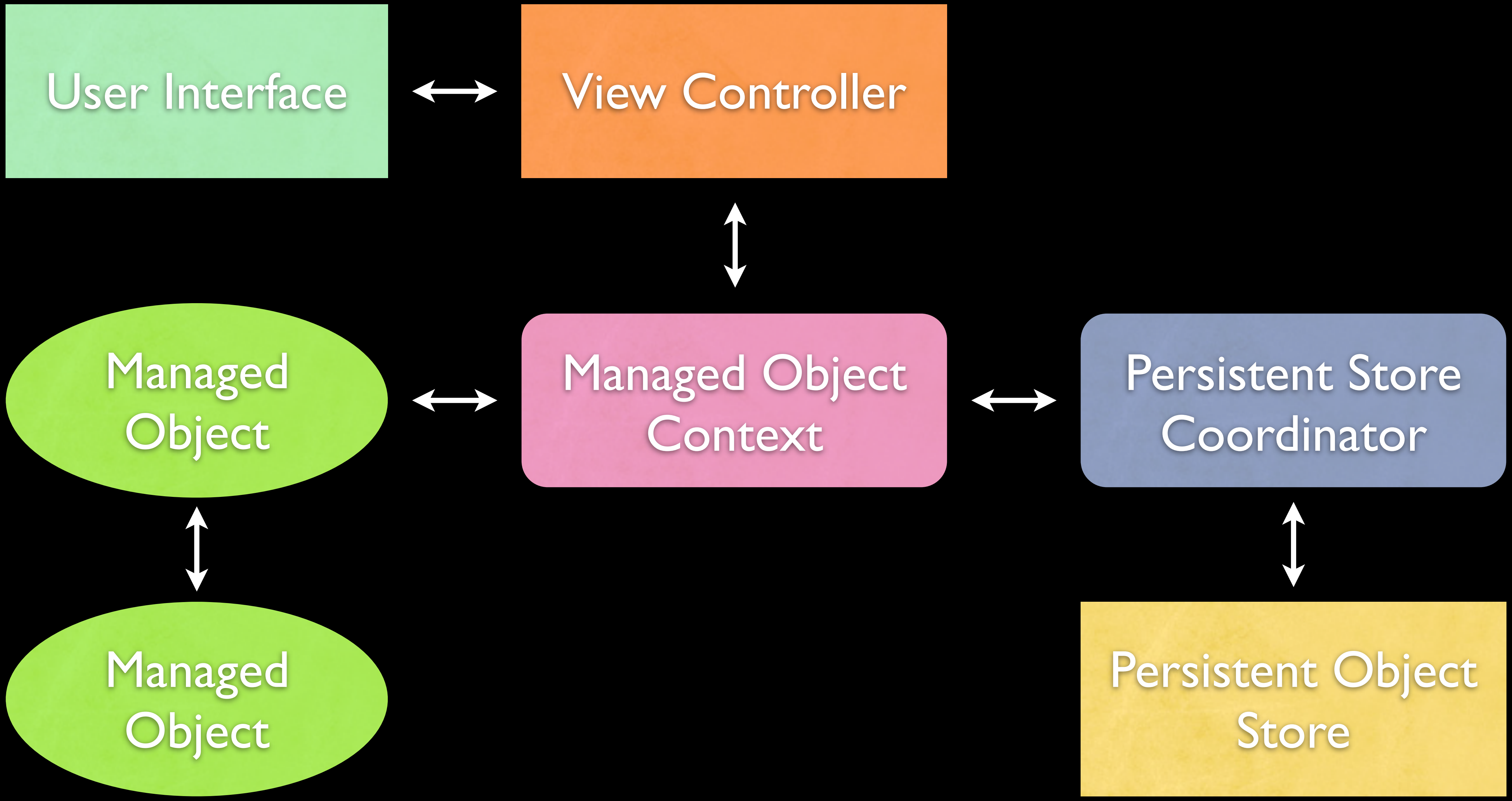
When to use it

- Large dataset (especially on iOS)
- Complex object graph
- Relationship management
- Property validation
- Complex custom document type
- 50-70% less model code

Basic parts

- Persistent store & coordinator
  - Filesystem coordination
- Managed object context(s)
  - “Scratch pads”
- Managed objects
  - Instances of data objects

Simplest form



User Interface

View Controller

Managed Object

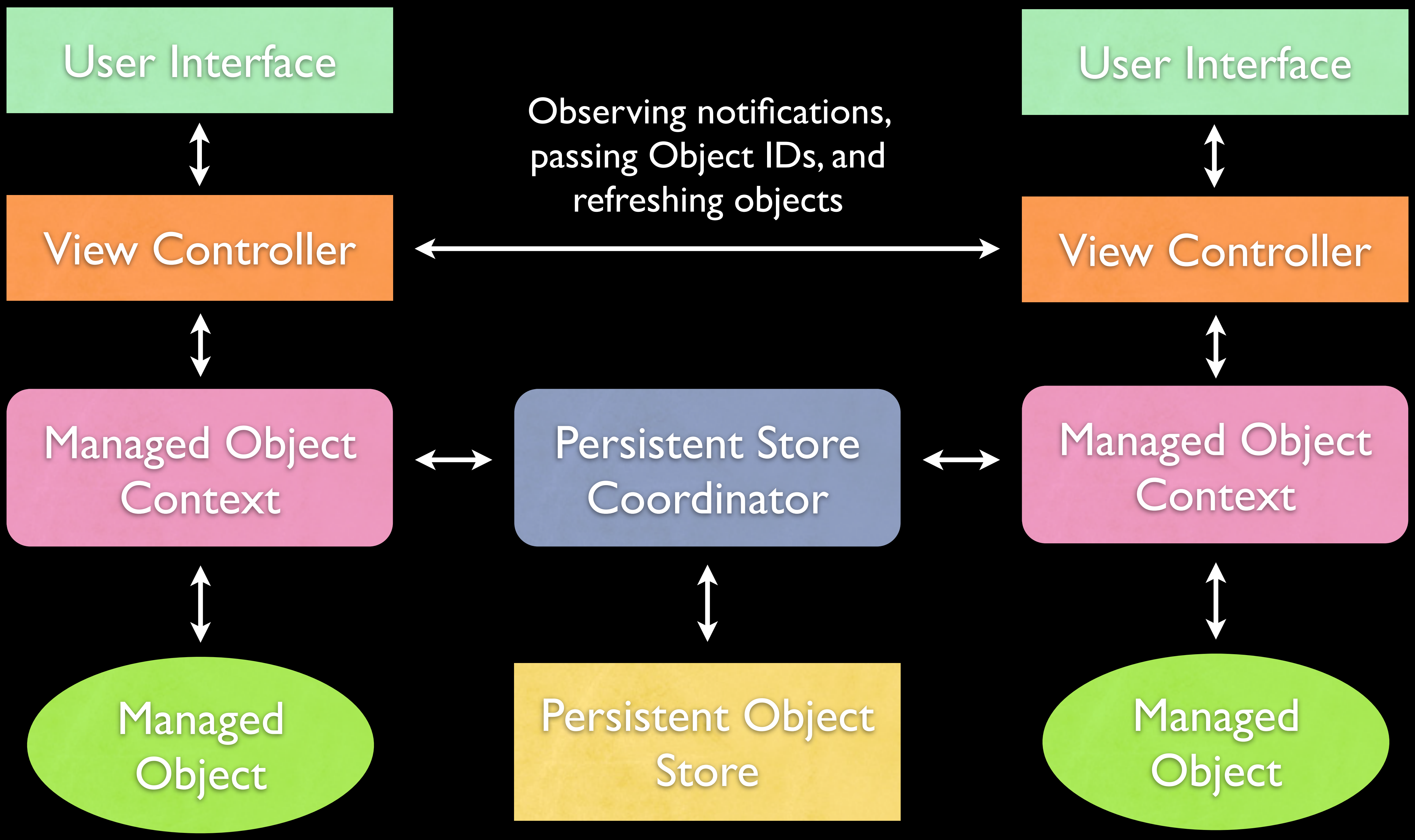
Managed Object Context

Persistent Store Coordinator

Managed Object

Persistent Object Store

More common form



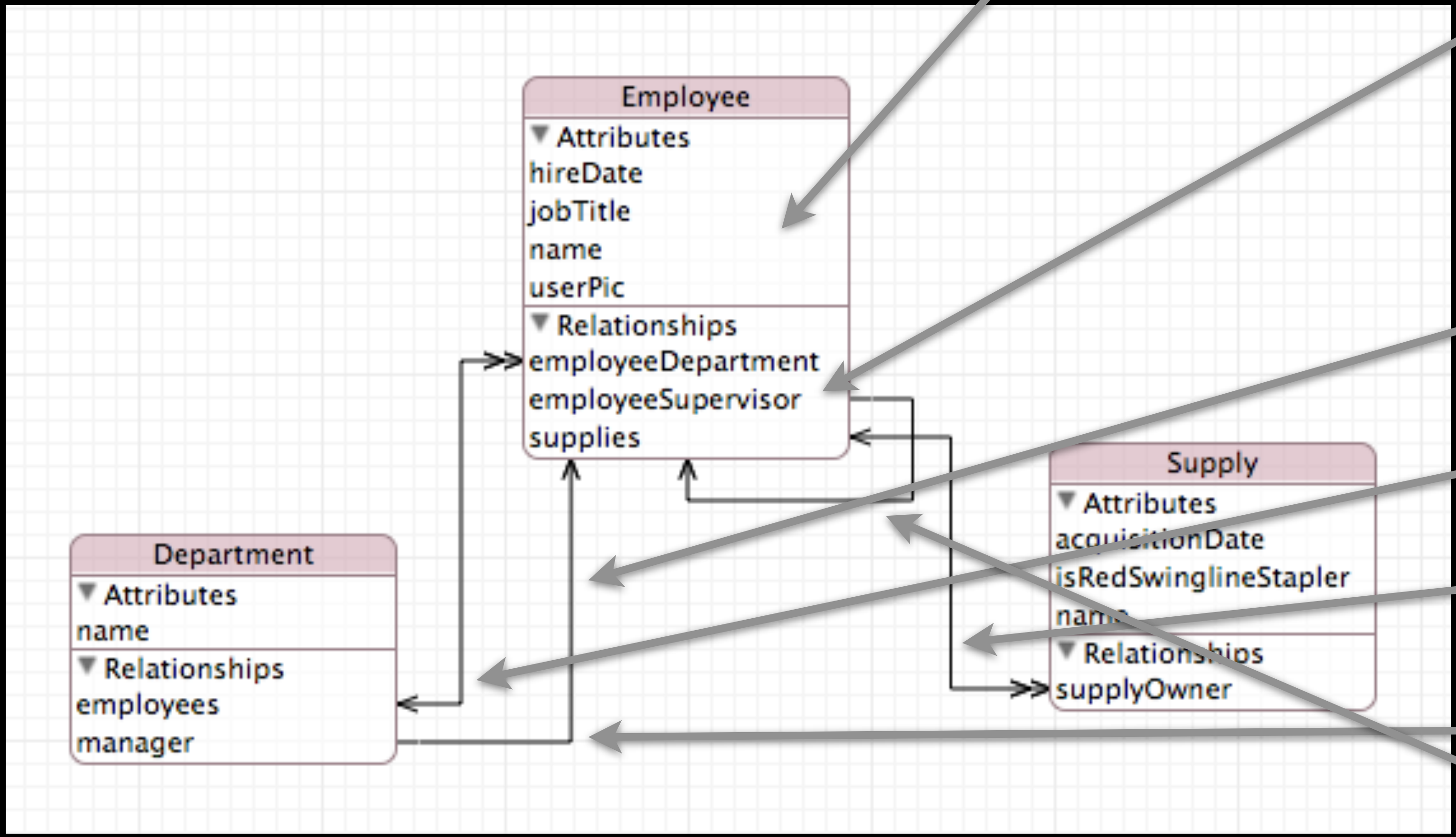
Persistent store types

Type	Platform	Speed	Graph
XML	Mac	slow	full
Atomic	both	fast	full
SQLite	both	fast	partial
In-memory	both	fast	full

Modeling your data

- Managed objects have:
  - An entity (like a class)
  - Attributes (like scalars)
  - Relationships (like object pointers)

Schema Example



Attributes

Relationships

To-one

To-many

Two-way

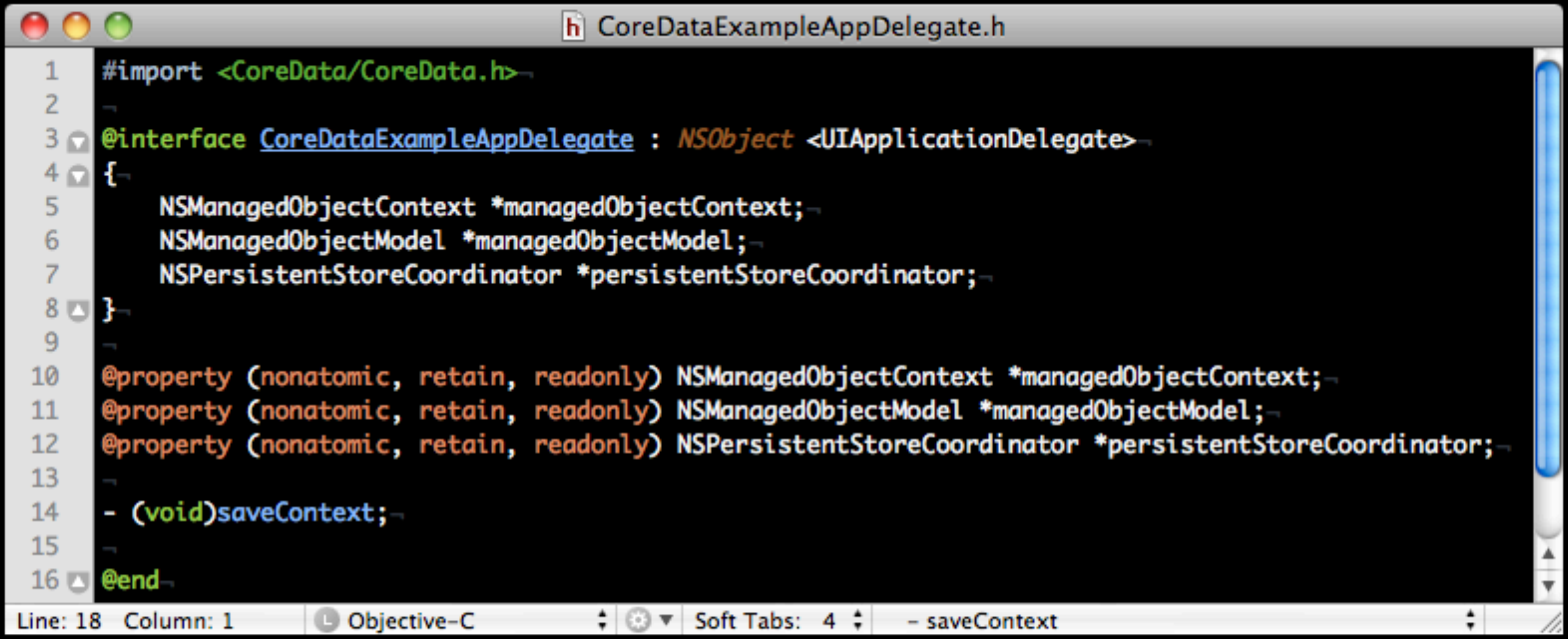
One-way

Same entity

Basic workflow

- Setup a managed object context
  - Backed by a persistent store coordinator
- Insert, delete, and fetch managed objects
- Save context changes
- Get notified of stuff

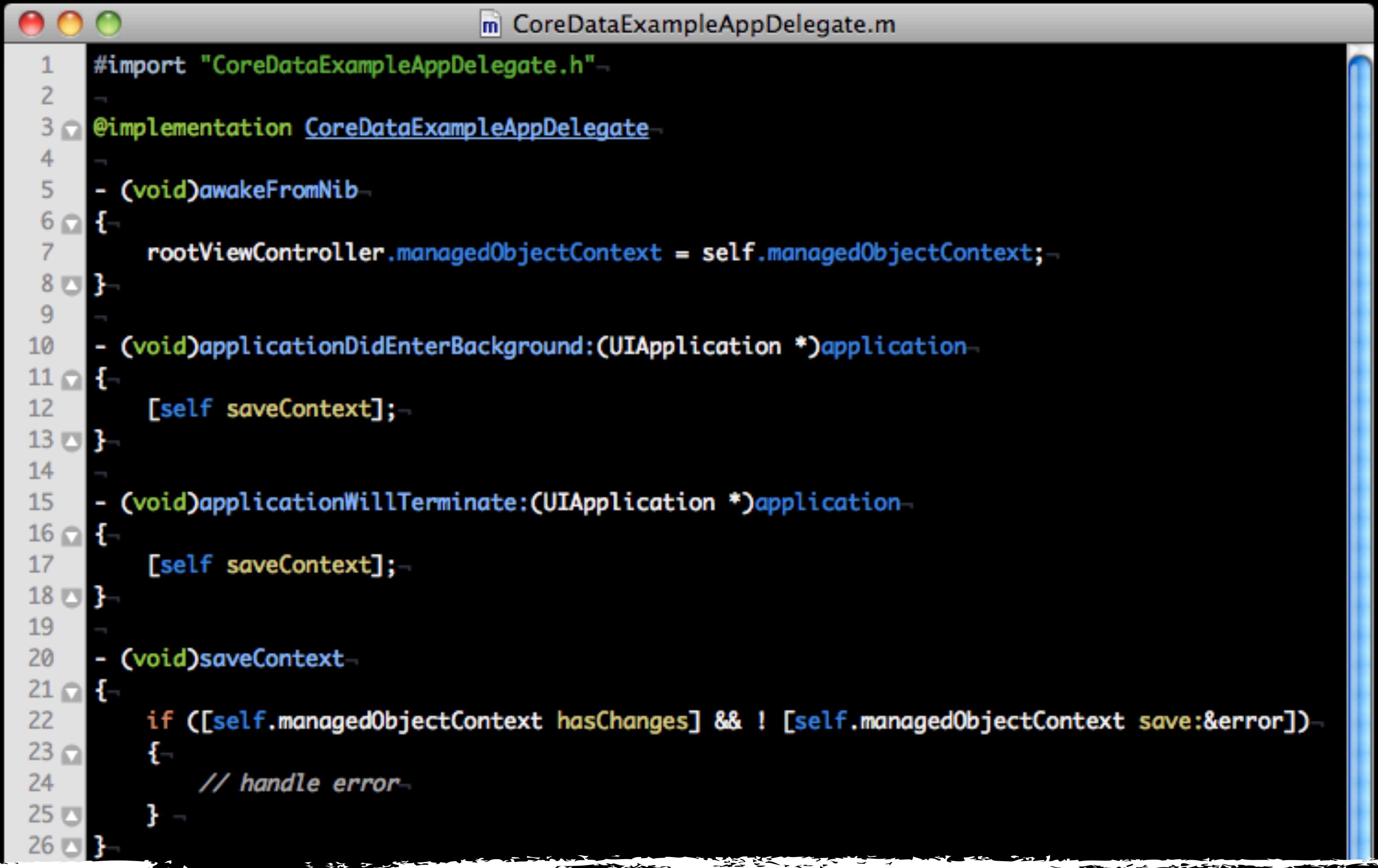
Setup context & store



```
#import <CoreData/CoreData.h>

@interface CoreDataExampleAppDelegate : NSObject <UIApplicationDelegate>
{
    NSManagedObjectContext *managedObjectContext;
    NSManagedObjectModel *managedObjectModel;
    NSPersistentStoreCoordinator *persistentStoreCoordinator;
}

@property (nonatomic, retain, readonly) NSManagedObjectContext *managedObjectContext;
@property (nonatomic, retain, readonly) NSManagedObjectModel *managedObjectModel;
@property (nonatomic, retain, readonly) NSPersistentStoreCoordinator *persistentStoreCoordinator;

- (void)saveContext;

@end
```

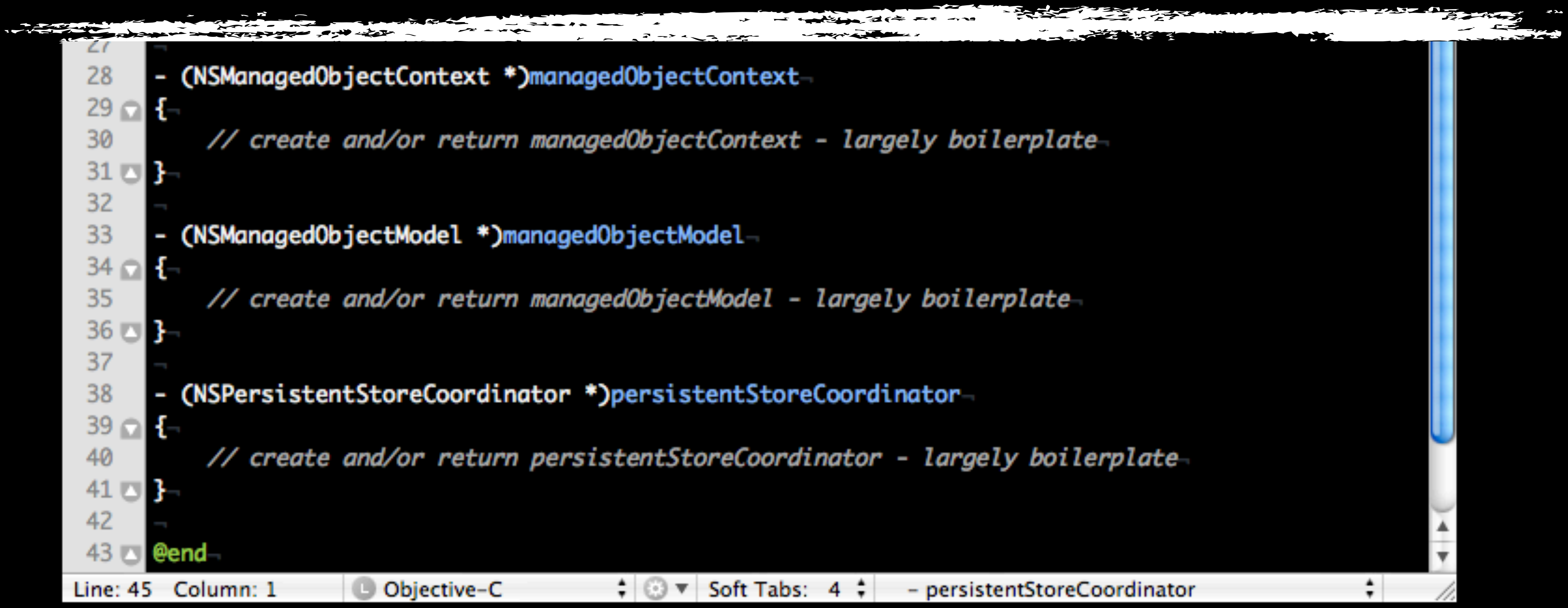
Line: 18 Column: 1 | Objective-C | Soft Tabs: 4 | – saveContext

Setup context & store



```
#import "CoreDataExampleAppDelegate.h"

@implementation CoreDataExampleAppDelegate

- (void)awakeFromNib
{
    rootViewController.managedObjectContext = self.managedObjectContext;
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self saveContext];
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    [self saveContext];
}

- (void)saveContext
{
    if ([self.managedObjectContext hasChanges] && ! [self.managedObjectContext save:&error])
    {
        // handle error
    }
}
```

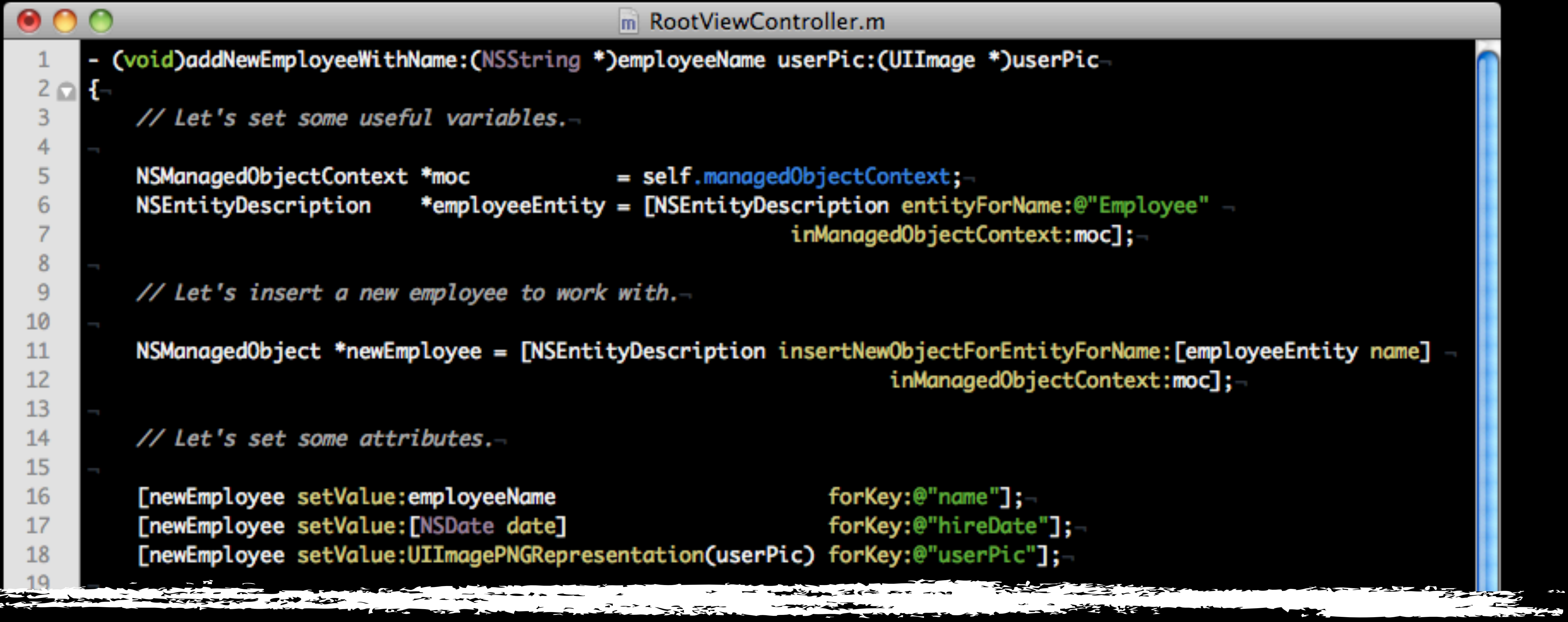
Setup context & store



```
- (NSManagedObjectContext *)managedObjectContext
{
    // create and/or return managedObjectContext - largely boilerplate
}

- (NSManagedObjectModel *)managedObjectModel
{
    // create and/or return managedObjectModel - largely boilerplate
}

- (NSPersistentStoreCoordinator *)persistentStoreCoordinator
{
    // create and/or return persistentStoreCoordinator - largely boilerplate
}

@end
```

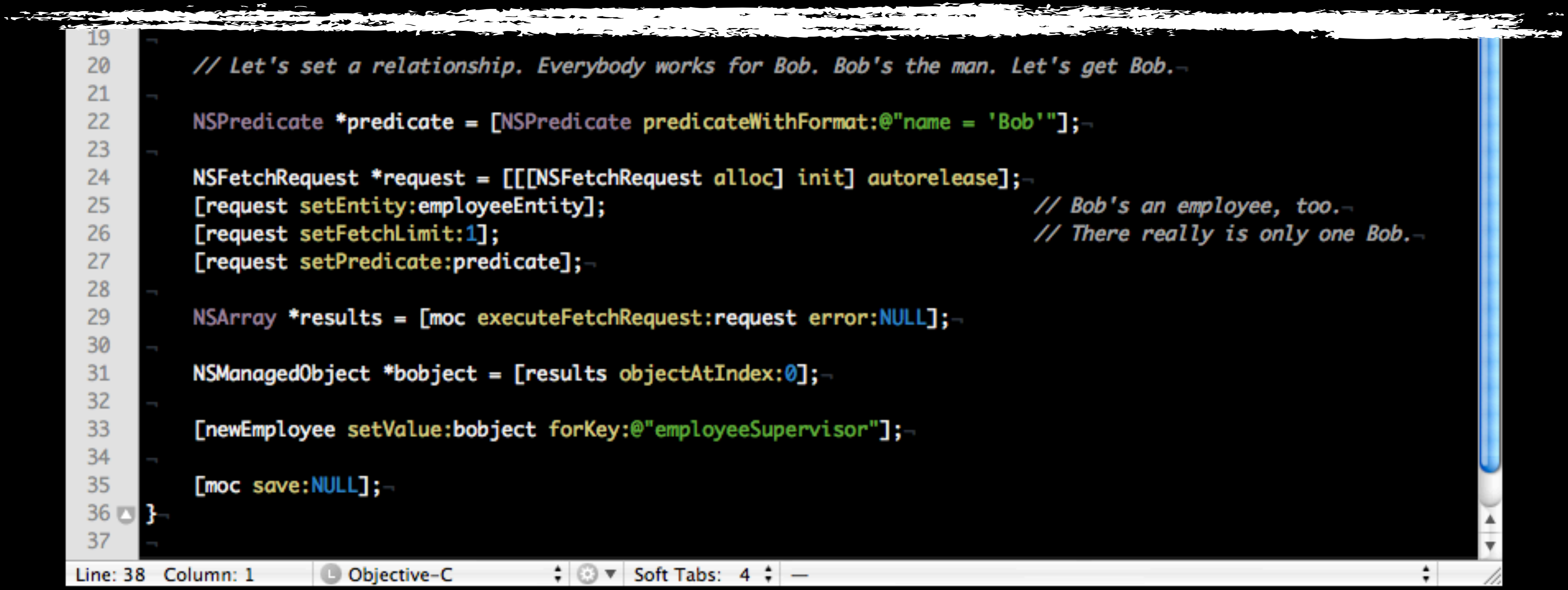
Line: 45 Column: 1 | Objective-C | Soft Tabs: 4 | - persistentStoreCoordinator

Manage objects



```
- (void)addNewEmployeeWithName:(NSString *)employeeName userPic:(UIImage *)userPic
{
    // Let's set some useful variables.

    NSManagedObjectContext *moc            = self.managedObjectContext;
    NSEntityDescription    *employeeEntity = [NSEntityDescription entityForName:@"Employee"
                                                         inManagedObjectContext:moc];

    // Let's insert a new employee to work with.

    NSManagedObject *newEmployee = [NSEntityDescription insertNewObjectForEntityForName:[employeeEntity name]
                                                                 inManagedObjectContext:moc];

    // Let's set some attributes.

    [newEmployee setValue:employeeName                      forKey:@"name"];
    [newEmployee setValue:[NSDate date]                     forKey:@"hireDate"];
    [newEmployee setValue:UIImagePNGRepresentation(userPic) forKey:@"userPic"];
```

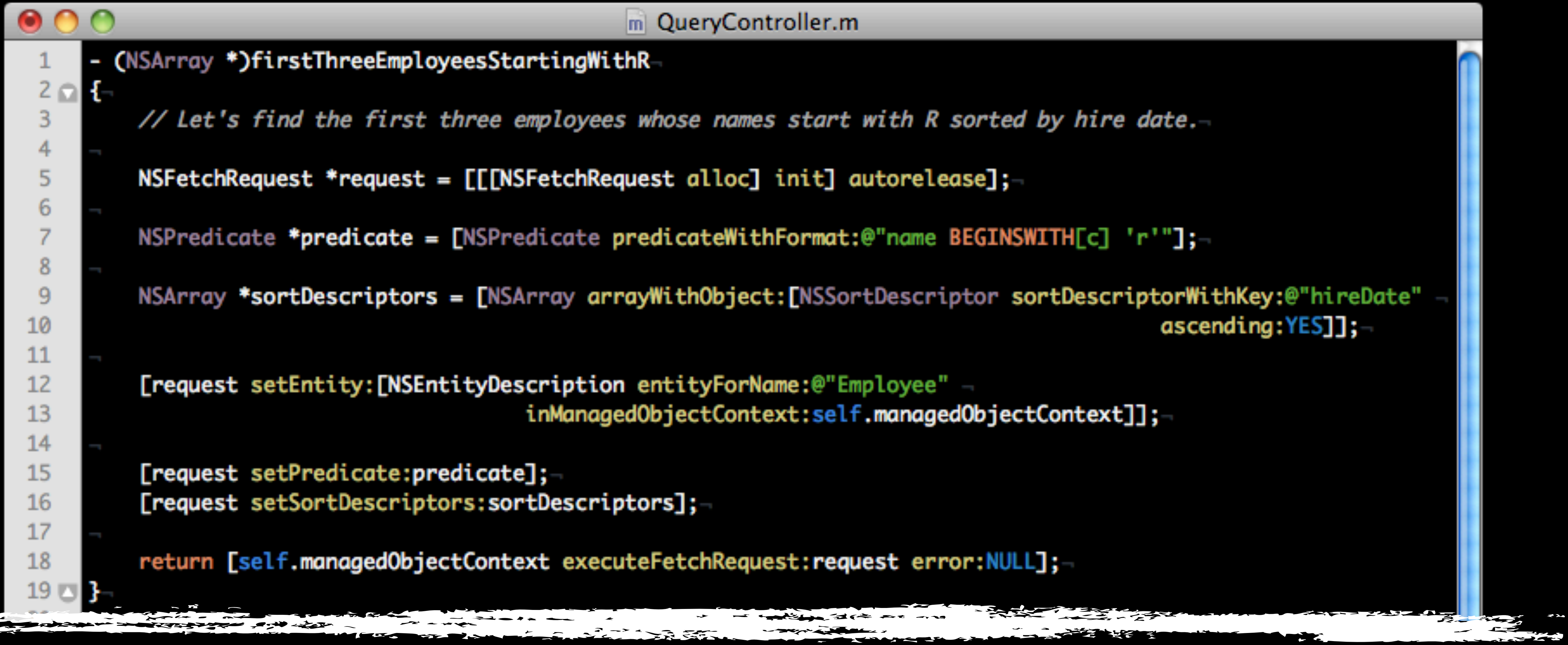
Manage objects



```objc

    // Let's set a relationship. Everybody works for Bob. Bob's the man. Let's get Bob.

    NSPredicate *predicate = [NSPredicate predicateWithFormat:@"name = 'Bob'"];

    NSFetchRequest *request = [[[NSFetchRequest alloc] init] autorelease];
    [request setEntity:employeeEntity];                                    // Bob's an employee, too.
    [request setFetchLimit:1];                                             // There really is only one Bob.
    [request setPredicate:predicate];

    NSArray *results = [moc executeFetchRequest:request error:NULL];

    NSManagedObject *bobject = [results objectAtIndex:0];

    [newEmployee setValue:bobject forKey:@"employeeSupervisor"];

    [moc save:NULL];
}

```

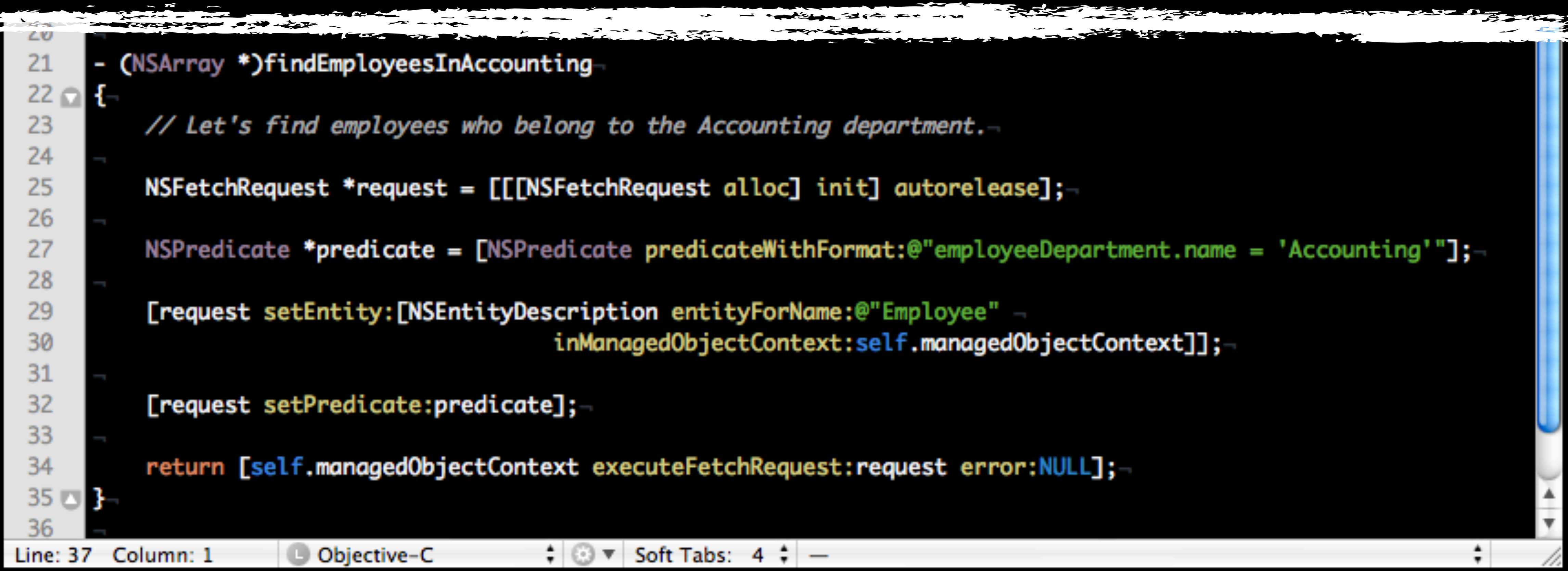
Line: 38 Column: 1 | Objective-C | Soft Tabs: 4 | —

Manage objects



```
- (NSArray *)firstThreeEmployeesStartingWithR
{
    // Let's find the first three employees whose names start with R sorted by hire date.

    NSFetchRequest *request = [[[NSFetchRequest alloc] init] autorelease];

    NSPredicate *predicate = [NSPredicate predicateWithFormat:@"name BEGINSWITH[c] 'r'"];

    NSArray *sortDescriptors = [NSArray arrayWithObject:[NSSortDescriptor sortDescriptorWithKey:@"hireDate"
                                                                                      ascending:YES]];

    [request setEntity:[NSEntityDescription entityForName:@"Employee"
                                   inManagedObjectContext:self.managedObjectContext]];

    [request setPredicate:predicate];
    [request setSortDescriptors:sortDescriptors];

    return [self.managedObjectContext executeFetchRequest:request error:NULL];
}
```

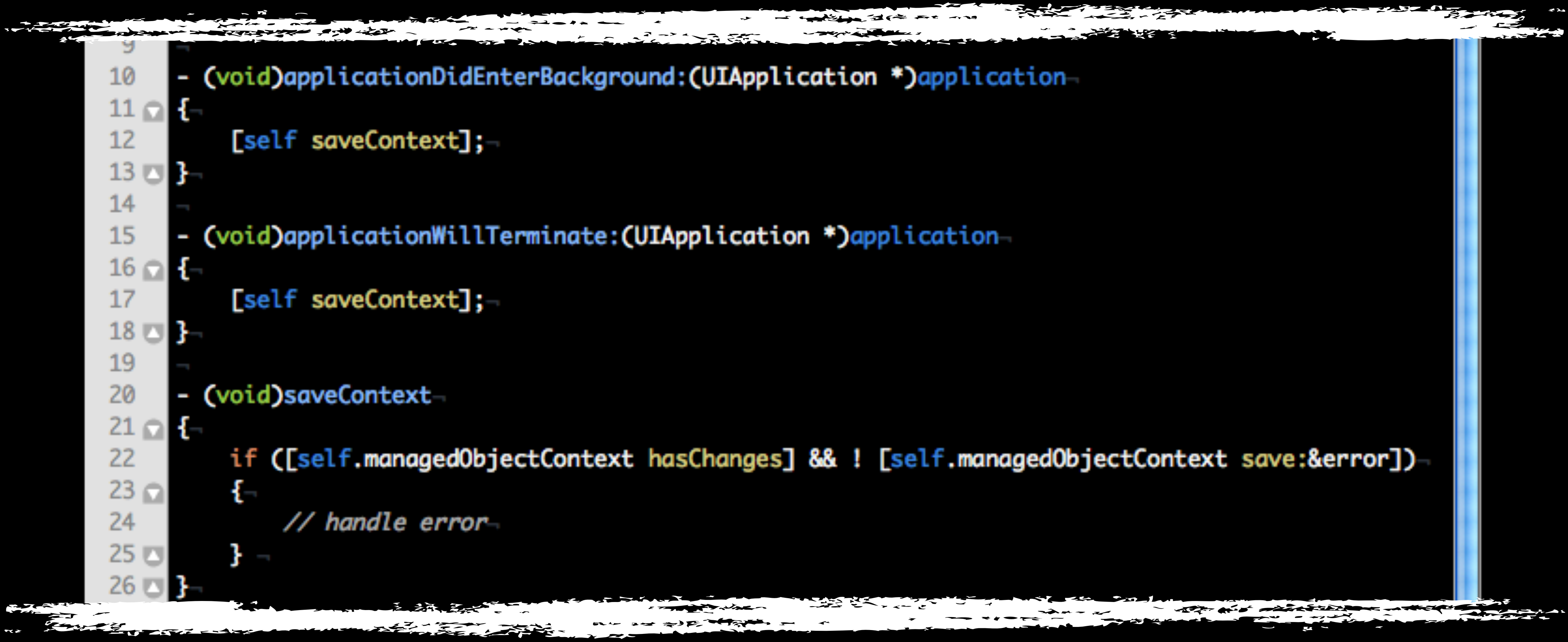
Manage objects



```
- (NSArray *)findEmployeesInAccounting
{
    // Let's find employees who belong to the Accounting department.

    NSFetchRequest *request = [[[NSFetchRequest alloc] init] autorelease];

    NSPredicate *predicate = [NSPredicate predicateWithFormat:@"employeeDepartment.name = 'Accounting'"];

    [request setEntity:[NSEntityDescription entityForName:@"Employee"
                                   inManagedObjectContext:self.managedObjectContext]];

    [request setPredicate:predicate];

    return [self.managedObjectContext executeFetchRequest:request error:NULL];
}

```

Line: 37 Column: 1 Objective-C Soft Tabs: 4 —

Save changes



```
- (void)applicationDidEnterBackground:(UIApplication *)application
{
    [self saveContext];
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    [self saveContext];
}

- (void)saveContext
{
    if ([self.managedObjectContext hasChanges] && ! [self.managedObjectContext save:&error])
    {
        // handle error
    }
}
```

Get notified

- `NSManagedObjectContext` notifications
  - `NSManagedObjectContextObjectsDidChangeNotification`
  - `NSManagedObjectContextDidSaveNotification`
  - `NSManagedObjectContextWillSaveNotification`
- Be sure to pass only `NSManagedObjectID`’s between contexts
- Generally, only save from the main thread

Quick Demo

Shout outs

- Faulting
- Fetched properties
- Undo management
- Concurrency
- Versioning & migration
- Bindings & `NSFetchedResultsController`
- mogenerator, emogenerator, et al.

Resources

- Core Data Programming Guide (*Apple*)
- Zarra book (general, *Pragmatic*)
- Tim & Tom tome (iOS, *Addison-Wesley*)