

Core Data Fundamentals

Beyond the Basics: More Reading and Writing

Brice Wilson
www.BriceWilson.net
[@brice_wilson](https://twitter.com/brice_wilson)



pluralsight 
hardcore developer training

Introduction

Custom Entity Classes

Entity Initialization


Reading and Writing Relationships

Query with NSPredicate

Custom NSManagedObject Subclasses

Entity

Name

Class 

☐ Abstract Entity


Parent Entity

Indexes

+ -

Entity

Name

Class 

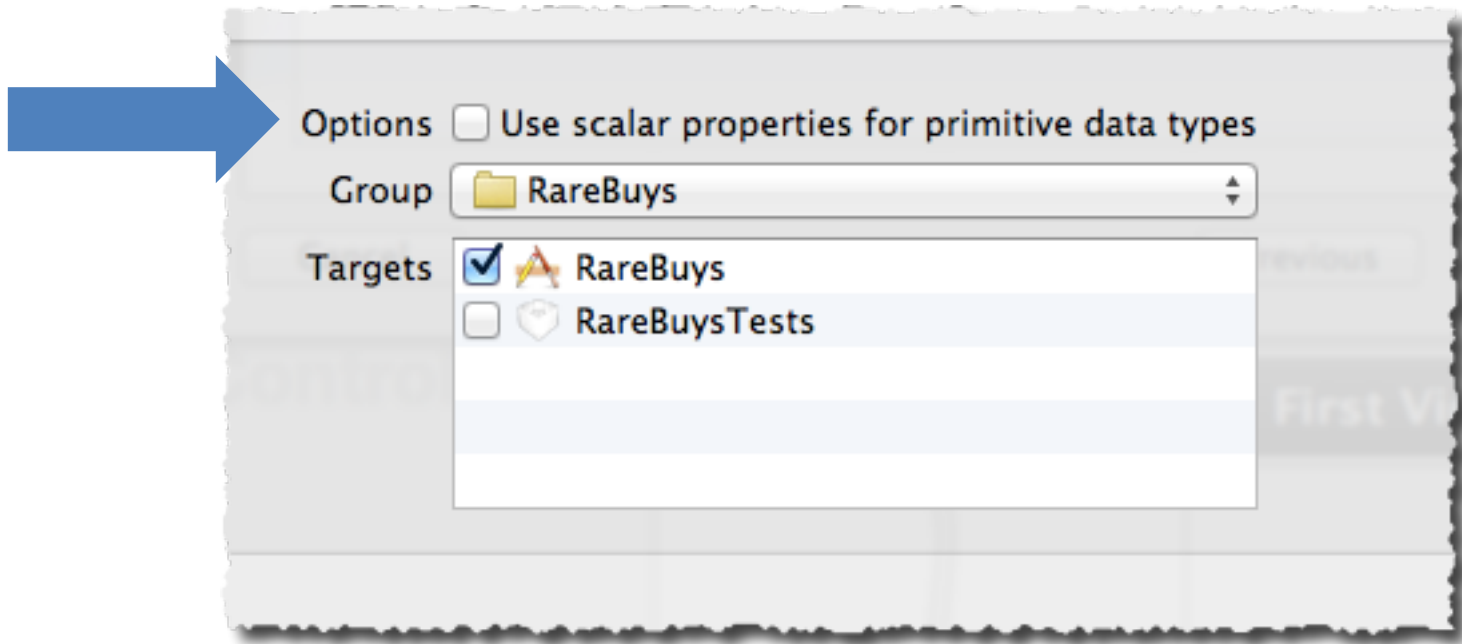
☐ Abstract Entity

Parent Entity

Indexes

+ -

Properties for Primitive Types



NSManagedObject Subclass Header

```
@interface ItemCategory : NSManagedObject
```

```
@property (nonatomic, retain) NSString * categoryDesc;  
@property (nonatomic, retain) NSString * categoryName;  
@property (nonatomic, retain) NSSet *items;  
@end
```

```
@interface ItemCategory (CoreDataGeneratedAccessors)
```

```
- (void)addItemObject:(RareItem *)value;  
- (void)removeItemObject:(RareItem *)value;  
- (void)addItem:(NSSet *)values;  
- (void)removeItems:(NSSet *)values;
```

```
@end
```

NSManagedObject Subclass Implementation

```
@implementation ItemCategory
```

```
@dynamic categoryDesc;
```

```
@dynamic categoryName;
```

```
@dynamic items;
```

```
@end
```

Custom Behavior – DON'Ts

- **Don't override the following**

- `primitiveValueForKey:`
- `setPrimitiveValue: forKey:`
- `isEqual:`
- `hash`
- `superclass`
- `class`
- `self`
- `zone`
- `isProxy`
- `managedObjectContext`
- `entity`
- `description`

Custom Behavior - DOs

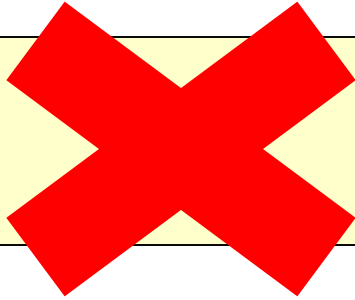
- **Do implement the following**
 - awakeFromInsert
 - awakeFromFetch

Managing To-One Relationships

```
NSObject *category = [item valueForKey:@"category"];
```

```
ItemCategory *category = [item category];
```

Managing To-Many Relationships



```
NSSet *stores = [item stores];  
[stores addObject:store];
```

```
NSMutableSet *stores = [item mutableSetValueForKey:@"stores"];  
[stores addObject:store];
```

```
[item addStoresObject:store];
```

NSPredicate

```
NSPredicate *pred = [NSPredicate predicateWithFormat:@"itemName CONTAINS %@", searchString];
```

```
NSPredicate *pred = [NSPredicate predicateWithFormat:@"%K CONTAINS %@", attr, searchString];
```

▪ String Comparison

- CONTAINS
- BEGINSWITH
- ENDSWITH
- LIKE
- MATCHES

▪ Basic Comparison

- =, ==
- >=, =>
- <=, =<
- <
- >
- !=, <>
- BETWEEN

▪ Aggregates

- ANY, SOME
- ALL
- NONE
- IN

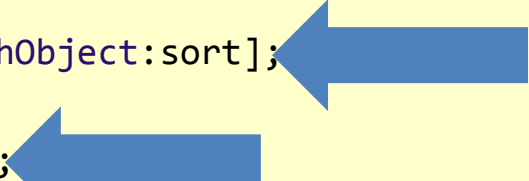
NSSortDescriptor

```
NSFetchRequest *request = [NSFetchRequest fetchRequestWithEntityName:@"RareItem"];

NSSortDescriptor *sort = [NSSortDescriptor sortDescriptorWithKey:@"itemName"
                                                                ascending:YES];

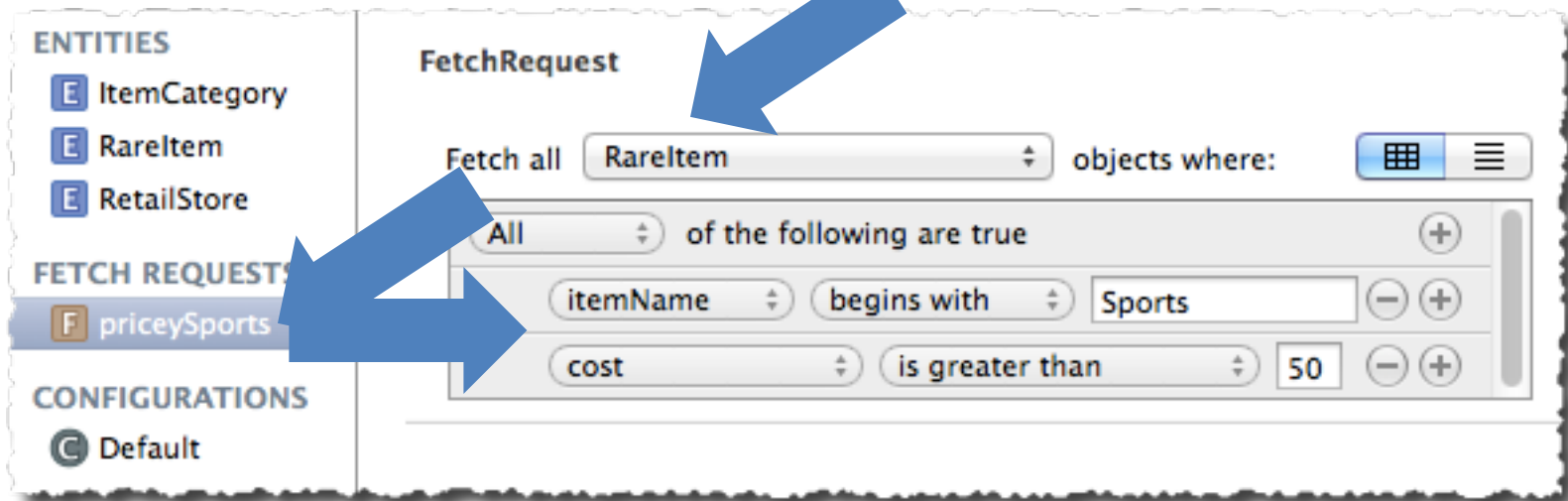
NSArray *sortArray = [NSArray arrayWithObject:sort];

[request setSortDescriptors:sortArray];
```



The diagram consists of two blue arrows. The first arrow originates from the 'sort' parameter in the 'arrayWithObject:sort' call of the third line and points to the 'sortArray' parameter in the 'setSortDescriptors:sortArray' call of the fourth line. The second arrow originates from the 'sortArray' parameter in the same 'setSortDescriptors:sortArray' call and points to the 'sortDescriptors' parameter in the same call.

Fetch Request Templates



Summary

Custom Entity Classes

Entity Initialization

Reading and Writing Relationships

Query with NSPredicate