Transformable Type in CoreData

What could possibly go wrong?

By Angelo Cammalleri
Very brief, what is CoreData and why?

- Data persistence solution by Apple.
- Abstraction of direct database handling.
- No SQL skill needed.
- Generates classes according to your CoreData model.
- Model contains relations and types of entities.
What is “Transformable” type in CoreData

“Boring: strings and integers; fun and mysterious: transformable!”

—Greg Heo
What is “Transformable” type in CoreData

- CoreData entities support String, Float, Boolean and Date as attributes.
- You can also use “Transformable”.
- Which means using your custom type.
What is “Transformable” type in CoreData

- Saving your NSManagedObject with more complex properties to CoreData.

- Instead of translating properties to aforementioned types.

- The same inverted for loading.
Sounds neat, how to use it?

- Our custom type must conform to NSCoding.
- Means NSArray, NSDictionary and NSData support out of the box!
- But please don’t...
What is happening?

- Supposedly minor update.
- Data loss report from testers…
- Not reproducible at first.
- Configuration: Dev != Test.
What is happening?

```swift
NSString *storeType = EWConfiguration.isDebug ? NSSQLiteStoreType : NSBinaryStoreType;
```

- With binary store unable to decode…
- …because CoreData lost its information of our transformable type.
What is happening?

- NSSecureCoding was introduced with iOS 11.
- No one at CoreData team knew about this in time.
- Would not lead to problems unless…
- …used with binary data store.
That must be the fix!

- A workaround found in the depths of Apple Developer Forums.
- Additional option `NSBinaryStoreSecureDecodingClasses`.
- CoreData can decode its data store again.
- But please don’t...

```swift
/**
 * Starting with iOS 11.0 the transformable type in CoreData is broken
 * But there is a workaround provided by Apple.
 * You can read more about that here: https://forums.developer.apple.com/thread/88194
 */
NSDictionary *options = @{
    NSMigratePersistentStoresAutomaticallyOption: @YES,
    NSInferMappingModelAutomaticallyOption: @YES,
    NSBinaryStoreSecureDecodingClasses: [NSSet setWithObjects:@"YourTransformable class", nil]
};

NSPersistentStore *persistenceStore = [persistentStoreCoordinator addPersistentStoreWithType: storeType
    configuration: nil
    URL: persistenceFile
    options: options
    error: error];
```
Not again!

- Supposedly minor update.
- Data loss report from **customers**…
- Reproducible…
Not again!

❖ Our transformable class was written in Swift.

❖ And moved from one module to another, thus changing the full class name…

❖ Now CoreData can’t find our transformable class.

```markdown
/**
  * Fix missing SendBoardingPassRestrictionModel due to migrating from framework to main app.
  * Causing data los for existing users.
  * More Information on this here: [https://stackoverflow.com/a/45290402/5097293](https://stackoverflow.com/a/45290402/5097293)
  */

[NSKeyedUnarchiver setClass: SendBoardingPassRestrictionsModel class forClassName: EWModel.SendBoardingPassRestrictionsModel];
```
What now?

- New regression tests.
- Removing transformables.
- Moving away from binary store type.
Sources

- https://gregheo.com/blog/core-data-transformable/