Working with Data Objects

The ability to easily interact with data in a database ranks high among the features that drive the success of most new programming languages and frameworks. This chapter talks about working with the data objects you create and store in your Core Data models. You’ll find Core Data provides an abstraction from the complexities of Structured Query Language (SQL) and database programming that allows you to quickly and efficiently read and write data to your persistent stores.

Understanding CRUD

Whether you claim the $R$ stands for Read or Retrieve and whether you side with Destroy over Delete for the $D$, we all agree that the acronym CRUD describes the four operations you perform on a persistent data store:

- Create
- Retrieve
- Update
- Delete

These four operations apply to all persistent store interaction, whether you’re working with Core Data in an iOS application, an Oracle database in a Java Enterprise Web application, a Virtual Storage Access Method (VSAM) file in a COBOL program, or any other situation in which you have a program that works with data. In fact, the CRUD concept has been extended to describe other situations in which you create, retrieve, update, and delete data. For example, the latest approach to providing services over the Web, Representation State Transfer (REST), has been called CRUD for the Web—the HTTP verbs map to the CRUD operations like this:
POST = Create
GET = Retrieve
PUT = Update
DELETE = Delete

In this chapter, you’ll build one application twice: once working directly with NSManagedObject instances and the second time with custom classes that extend NSManagedObject. This section builds the raw NSManagedObject version. With a nod to Scott Hanselman’s Baby Smash! (www.hanselman.com/babysmash/), this application, called Shapes, has the following requirements and characteristics:

- Each time someone taps the screen, a random shape appears where the screen was tapped.
- The shape is randomly a circle or a polygon.
- If a polygon, the shape has a random number of sides and can be concave, convex, or mixed.
- If a circle, the shape has a random radius.
- Shapes appear in random colors.
- Shaking the device deletes all the shapes.
- Rotating the device updates all the shapes to random colors.
- The screen splits down the middle, and each shape appears twice, once on each screen half.
- One of the screen halves is zoomed at 2x magnification (meaning some shapes won’t appear, because they’ll fall outside the screen’s boundaries).
- Shapes is built for the iPad to take advantage of the extra screen space.

From a Core Data perspective, Shapes illustrates the following:

- **Create**: Each time you tap the screen, Shapes creates a shape object in the persistent store.
- **Retrieve**: Each time the screen draws the shapes, Shapes retrieves the shapes from the persistent store.
- **Update**: Each time you rotate the device, Shapes updates all the shapes in the persistent store with different random colors.
- **Delete**: Each time you shake the device, Shapes deletes all the shapes from the persistent store.
- **Inheritance**: The Polygon and Circle entities inherit from a common parent, Shape.