

# Saving Data in iOS

Hands-On Challenges

# Saving Data in iOS Hands-On Challenges

Copyright © 2014 Razeware LLC.

All rights reserved. No part of this book or corresponding materials (such as text, images, or source code) may be reproduced or distributed by any means without prior written permission of the copyright owner.

This book and all corresponding materials (such as source code) are provided on an "as is" basis, without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

All trademarks and registered trademarks appearing in this book are the property of their respective owners.

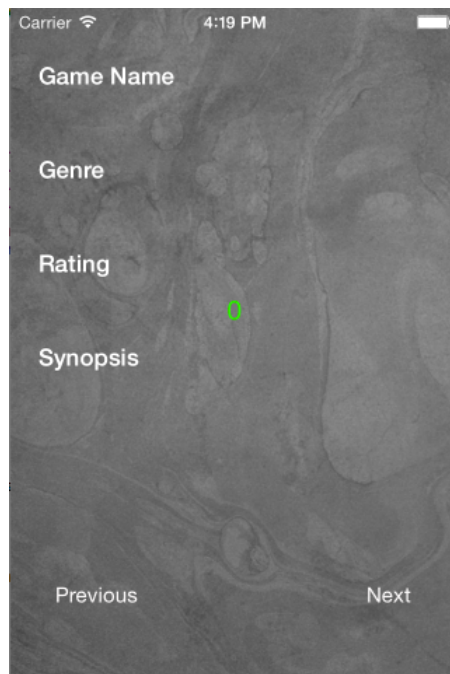


# Playing with Property Lists

Property lists are a very versatile tool that you'll be using throughout your iOS development career. They make for a great way to add runtime configuration to your application, but they also provide a way to model your objects. In this upcoming challenge, you will be creating objects from a property list.

## Challenge: Creating Objects

Open the sample application and build and run. You should see the following in the simulator.



This is the empty template of a video game review app. Granted, it won't win you any design awards, it will show you how to use property lists.

Open **ViewController.m** and you will see that it contains a few methods. The `previousTap:` and `nextTap:` allows you to navigate through the various reviews. The `updateUI:` displays the current review.

Each review is represented by a `RWTVideoGame` class. It's a simple class that contains properties for name, genre, rating, and the synopsis. All these objects are contained in an array, and as the user cycles through the different reviews, the



view controller grabs the appropriate review from the array, then updates the user interface.

Open **reviews.plist**. You will see that it contains an array of dictionaries. These dictionaries encapsulate the review data. Each of the property fields correspond to class properties.

Now open **ViewController.m**. At the top of **viewDidLoad:**, add the following code to get a path to the property list:

```
NSString * properListPath = [[NSBundle mainBundle]
    pathForResource:@"reviews" ofType:@"plist"];
```

Whenever you need to access elements inside of your bundle, `pathForResource:` will return a string path for the actual resource. You can also use `urlForResource:` as well. With a reference to the path, you can now read the property list.

Your challenge is to read the property list, create all the objects, and assign them to `videoGames` property. Once you build and run, you should then see all the various reviews.

## Uber Haxx0r Challenge: Importing New Episodes

Reading a property list from a bundle is quite useful for application settings, but as a way to manage your objects, it can be limited. Bundles are read only so if your objects change, you have no way to write them back to the bundle. In this case, you'll want to write them to a directory.

This is the Uber Haxx0r challenge. You first must read from a property list, then once you have created all the objects, you should write them out to a property list in the Library directory.

Once you have written out to the Library directory, the app should use that property list going forward. If that list is missing, then the app should default to using the property list included with the bundle.

Good luck!

