**Topic: Settings**   
(Previous topic: table views)

Announcements

* Term project …

Intro

* Show a few apps and how their settings are done in the Settings app
* Demo the Bridge Control app and show it’s settings

Settings Bundle

* A settings bundle is a group of files built in to an application that tells the Settings application which preferences the application wishes to collect from the user.
* Each settings bundle must contain a property list called Root.plist that defines the root-level preferences view.
* Settings.bundle. Expand the Settings. bundle item, and you should see two subitems:
  + a folder named en.lproj, containing a file named Root.strings
  + another named Root.plist.
* Property lists are essentially dictionaries. Use key-value pairs
  + Nodes can be simple data types, or arrays or dictionaries
  + Item 0 in the PreferenceSpecifiers array in a settings bundle property list should always be a PSGroupSpecifier , so that the settings start in a new group. (This is the group for your app that will be displayed in the Settings app)
    - The values in parentheses represent the value of the Type item (the first Group ) and the Title item (the second Group ).

Reading Settings in the Application

* We’ll use a class called NSUserDefaults to access the user’s settings. NSUserDefaults is implemented as a singleton, which means there is only one instance of NSUserDefaults running in our application. To get access to that one instance, we call the class method standardUserDefaults , like so:  
   *let defaults = NSUserDefaults.standardUserDefaults()*
* use it much like a Dictionary . To get a value from it, we can call objectForKey: , which will return an object, a String or a Foundation object such as NSDate , or NSNumber . If we want to retrieve the value as a scalar—like an int , float ,or Boolean —we can use another method, such as intForKey() , floatForKey() , or boolForKey() . For example:  
   *officerLabel.text = defaults.stringForKey(“officer”)*
* Some specifiers were used to interface objects for user interaction. Those are the specifiers we are really interested in because they hold the keys the real settings data.

Refreshing the settings in the application

hitting the Home button while an app is running doesn’t actually quit the app.

Instead, the operating system suspends the app in the background