iOS App Development

2.5 Local Data Management

Local Data Management

- Preferences and Settings
  - NSUserDefaults
- File System Structure
- Property List
- Object Archiving
- Advanced Topics

Preference and Settings

- User Defaults : NSUserDefaults class
- ~~Preferences in iCloud: iCloud Key-Value Store~~
- iOS Settings Bundle

User Defaults System

- Preference
  - Domain: App specific | Systemwide
    - Name Key
    - Value (NSData, NSString, NSNumber, NSDate, NSArray, or NSDictionary)
- NSUserDefaults class to access App's preferences

NSUserDefaults class

- App domain preferences
- Get/Set/Remove preference values
- +(NSUserDefaults *) standardUserDefaults method of NSUserDefaults class for the single/shared object

Default Preferences

- No preference value set? `0` or `nil` by default
- NSRegistrationDomaion

```
// UIApplicationDelegate subclass
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Register the preference defaults early.
    NSDictionary *appDefaults =
[NSDictionary dictionaryWithObject:[NSNumber numberWithBool:YES]
                                    forKey:@"cacheData"];
    [[NSUserDefaults standardUserDefaults] registerDefaults:appDefaults];

    // Other initialization...

    return YES;
}
```

Getting Preference Values

- NSUserDefaults
  - - arrayForKey:
  - - dictionaryForKey:
  - - floatForKey:
  - - integerForKey:
  - - objectForKey: // id
  - - doubleForKey:
  - - URLForKey: // NSURL *

```
// Example
if ([[NSUserDefaults standardUserDefaults] boolForKey:@"cacheData"]) {
        // Delete the backup file.
}

NSUserDefaults *defaults = [NSUserDefaults standardUserDefaults];
NSURL *defaultImageURL = [defaults URLForKey:@”defaultImageURL”];
...
```

Setting Preference Values

- setBool:forKey:
- setFloat:forKey:

```
// Example
NSUserDefaults* defaults = [NSUserDefaults standardUserDefaults];

[defaults setBool:YES forKey:@"cacheData"];
[defaults setObject:[NSDate dateWithTimeIntervalSinceNow:(3600 * 24 * 7)]
             forKey:@"CacheExpirationDate"]; // Set a 1-week expiration
    // or
[defaults setBool:NO forKey:@"cacheData"];
[defaults removeObjectForKey:@"CacheExpirationDate"]; // Removing a value!
```

- setObject:forKey: // Array, Dictionary, String, Data,...

Synchronizing Preference Changes

- - synchronize
  - Writes modifications
  - Reads changes (unmodified preferences only)
- User defaults database could be changed somewhere else, such as Settings app

Detecting Preference Changes

- NSUserDefaults object sends NSUserDefaultsDidChangeNotification
- Observe using NSNotificationCenter's

```
// Example
NSNotificationCenter *defaultCenter = [NSNotificationCenter defaultCenter];
[defaultCenter addObserver:self
                  selector:@selector(userDefaultsChanged:)
                      name:NSUserDefaultsDidChangeNotification
                    object:nil];
...

- (void) userDefaultsChanged:(NSNotification *) notification {
    // Preferences changed outside
    NSUserDefaults *defaults = (NSUserDefaults *) [notification object];
    NSLog(@"%@", [defaults objectForKey:@"concernedKey"]);
}
```

Preferences in iCloud

- Key-value store
- NSUbiquitousKeyValueStore object plays similar role as NSUserDefaults
- Intended for
  - Small amount of data: limited space per app: 64KB (iOS 5)
  - Synchronization between apps across iCloud

iOS Settings Bundle

- Use of Setting Bundle allows Settings app to manage preferences
- Use NSUserDefaults to access preferences defined by Setting Bundle
- We don’t cover this part in this session. Please refer to the documentation

File System

- Application Sandbox
  - Security/Privacy/Cleanup
    - Application Bundle Directory (read-only)
    - Documents Directory
    - Caches Directory

NSFileManager

- Generic File-system operations (except reading/writing)
- - (NSArray *)URLsforDirectory:inDomains:
  - To get the URL for the paths available to the App
  - NSSearchPathDirectory examples
    - NSDocumentDirectory
    - NSCachesDirectory
- NSString class
  - - stringByAppendingPathComponent:

NSBundle

- Accessing resources under Application Bundle directory
  - + (NSBundle *) mainBundle
  - + (NSBundle *) bundleWithURL:
  - + (NSBundle *) bundleWithPath:
  - - URLForResource:withExtension:
- See Bundle Programming Guide, Apple for more details

Property List

- NSArray/NSDictionary methods
  - - writeToURL:atomically: or - writeToFile:atomically:
  - - initWithContentsOfURL: or - initWithContentsOfFile:
- NSPropertyListSerialization class
  - Property List <-> NSData
  - + dataFromPropertyList:format:errorDescription:
  - + propertyListWithData:options:format:error:

Object Archiving

- NSCoding Protocol
  - - (void) encodeWithCoder:(NSCoder *)coder
  - - initWithCoder:(NSCoder *)coder;
- Object Graph -> Persistent Store:
  i.e. File

Advanced Topics

- Not covered today
  - Core Data Framework (sqlite)
    - Property List is not efficient for large amount of data records i.e. > 1000
  - Remote Data: Web Service over Internet

References

- Preferences and Settings Programming Guide, Apple
- Stanford University iTunesU Lecture 12, CS193p 2011 Winter