Persistant Storage

Storage

In a computer, storage is the place where data is held in an electromagnetic or optical form for access by a computer [processor](http://searchcio-midmarket.techtarget.com/definition/processor). There are two general usages.

1) [Storage](http://searchcio-midmarket.techtarget.com/definition/core-competency) is frequently used to mean the devices and data connected to the computer through input/output operations i.e, hard disk and tape systems and other forms of storage that don't include computer memory and other in-computer storage.

2) In a more formal usage, storage has been divided into:

(1) primary storage, which holds data in [memory](http://searchmobilecomputing.techtarget.com/definition/memory) (which is called random access memory) and other "built-in" devices such as the processor's cache, and

(2) [secondary storage](http://searchstorage.techtarget.com/definition/secondary-auxiliary-storage), which holds data on [hard disk](http://searchstorage.techtarget.com/definition/hard-disk)s, tapes, and other devices requiring input/output operations.

Persistance

**persistence** refers to the characteristic of [state](https://en.wikipedia.org/wiki/State_(computer_science)) that outlives the [process](https://en.wikipedia.org/wiki/Process_(computing)) that created it. This is achieved in practice by storing the state as data in [computer data storage](https://en.wikipedia.org/wiki/Computer_data_storage). Programmers have to transfer data to and from storage devices and have to provide mappings from the programming data structures to the storage device data structures via [serialization](https://en.wikipedia.org/wiki/Serialization) of data and [structure](https://en.wikipedia.org/wiki/Data_structure) to formats compatible with the storage.

For ex: Picture editing programs or word processors achieve [state](https://en.wikipedia.org/wiki/State_(computer_science)) persistence by saving their documents to files.

Persistant Techniques

1. System Images
2. Journals
3. Dirty Writes

* Using [system images](https://en.wikipedia.org/wiki/System_image) is the simplest persistence strategy. Notebook [hibernation](https://en.wikipedia.org/wiki/Hibernate_(OS_feature)) is an example of orthogonal persistence using a system image because it does not require any actions by the programs running on the machine.
* Using journals is the second simplest persistence technique. Journaling is the process of storing events in a log before each one is applied to a system. Such logs are called journals.
* On startup, the journal is read and each event is reapplied to the system, avoiding data loss in the case of system failure or shutdown.
* The entire "Undo/Redo" history of user commands in a picture editing program, for example, when written to a file, constitutes a journal capable of recovering the state of an edited pictures at any point in time.
* This technique is the writing to storage of only those portions of system state that have been modified since their last write. Sophisticated document editing applications, for example, will use dirty writes to save only those portions of a document that were actually changed since the last save.

Persistant Storage

It denotes a digital storage facility that doesn't lose its data after it loses its power supply. Persistent [storage](http://searchstorage.techtarget.com/definition/storage) is any [data](http://searchdatamanagement.techtarget.com/definition/data) storage device that retains data after power to that device is shut off. It is also sometimes referred to as [non-volatile storage](http://searchstorage.techtarget.com/definition/nonvolatile-storage).

Types of Persistant Storage

* Hard disk Drives
* Solid State Drives

This can be in the form of [file](http://searchstorage.techtarget.com/definition/file-storage), [block](http://searchsqlserver.techtarget.com/definition/block) or [object storage](http://searchstorage.techtarget.com/definition/object-storage). On the other hand, [RAM](http://searchstorage.techtarget.com/definition/RAM-random-access-memory) and [cache](http://searchstorage.techtarget.com/definition/cache-memory) are typically non-persistent, and are erased of data when power is turned off. However, certain types such as [non-volatile RAM](http://searchstorage.techtarget.com/definition/nonvolatile-memory) and flash-based RAM are persistent.

Persistence is beneficial so that in the event of a crash or reboot, data is not lost. It doesn’t loose any data after loses its power supply.

Data Persistance in IOS

There are several strategies for storing application data on disk. Four common approaches an ios.

* NsUserDefaults
* Property Lists
* SQLIte
* Core Data

Property Lists

Property List documents are an archived tree of objects. The Property List contains a root element which is either an NSDictionary or an NSArray. Inside the root element is the archived data. The classes that can be archived into a Property List are: NSString, NSNumber, NSData, NSDate, NSArray and NSDictionary. Providing any other objects to be archived as a property list will not write the file. A common mistake is to give a Boolean or Integer object to one of the items in the NSArray or NSDictionary. To store a Boolean or Integer use the NSNumber class.

SQLite

using the SQLite C library can be tricky the first time around as it is very particular about the argument types given to functions. It's important to remember the differences when passing in strings and the arguments required for these functions.

If you want to bypass the C library there are a few Objective-C wrappers written for using SQLite on the iOS. [FMDB](https://github.com/ccgus/fmdb) is one particular wrapper that you can include into your Xcode project to avoid using the C library.

Core Data

Core Data it is important to remember to create your Xcode project with the “Use Core Data for storage” option checked. That option will create some required properties and methods in your Application Delegate as well as include the Core Data framework into your project.

Core Data uses SQLite by default as the main database in your application. The Core Data framework in the iOS allows for 2 different database storage types and by default it is SQLite.

NSUserDefaults

It is not used for storing collections of data as demonstrated in the other 3 implementations. Information stored in this method should be limited to user changeable settings and options.

#### **Storing user authentication information**

When storing usernames and passwords it is better to use the Keychain service available in the iOS. The Keychain is a secure storage medium to keep user authentication information on the device in a secure and encrypted manner. For more information on the Keychain go to the [Keychain Services Programming Guide](https://developer.apple.com/library/ios/#documentation/Security/Conceptual/keychainServConcepts/01introduction/introduction.html)in the Apple Documentation.

With NSUserDefaults you can save objects from the following class types:

* NSData
* NSString
* NSNumber
* NSDate
* NSArray
* NSDictionary

If you want to store any other type of object, such as a UIImage, you will typically need to archive it or wrap it in an instance of NSData, NSNumber, or NSString.

Implementation of NSUserDefaults in xcode:

## **Saving**

NSUserDefaults \*prefs = [NSUserDefaults standardUserDefaults];

// saving an NSString

[prefs setObject:@"TextToSave" forKey:@"keyToLookupString"];

// saving an NSInteger

[prefs setInteger:42 forKey:@"integerKey"];

// saving a Double

[prefs setDouble:3.1415 forKey:@"doubleKey"];

// saving a Float

[prefs setFloat:1.2345678 forKey:@"floatKey"];

## **Retrieving**

NSUserDefaults \*prefs = [NSUserDefaults standardUserDefaults];

// getting an NSString

NSString \*myString = [prefs stringForKey:@"keyToLookupString"];

// getting an NSInteger

NSInteger myInt = [prefs integerForKey:@"integerKey"];

// getting an Float

float myFloat = [prefs floatForKey:@"floatKey"];

Benefits of DataPersistance

1. Data loss will be reduced
2. Provides security
3. Less time consumption

I