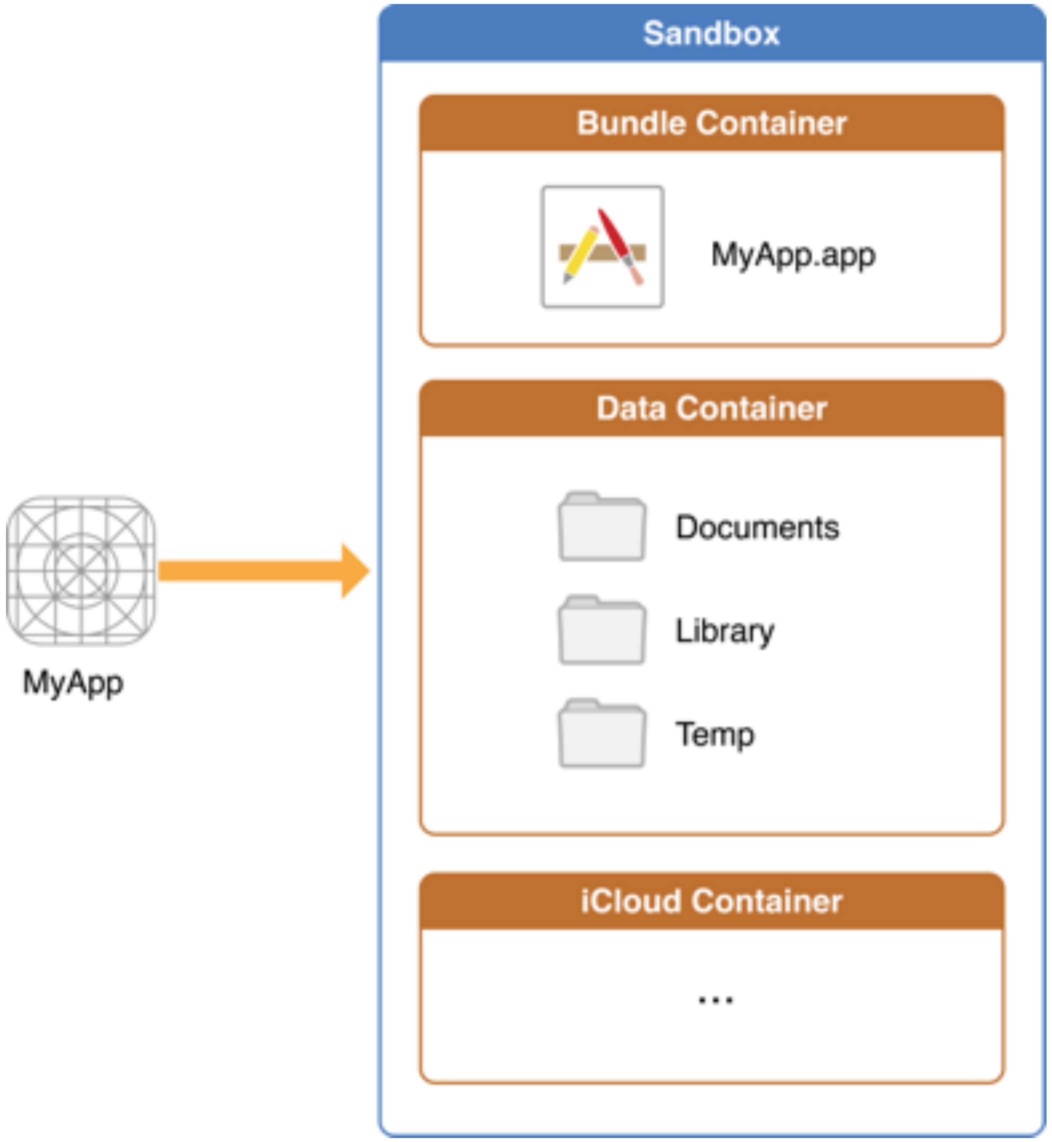
File System Basics

Lecture 5

File System Basics

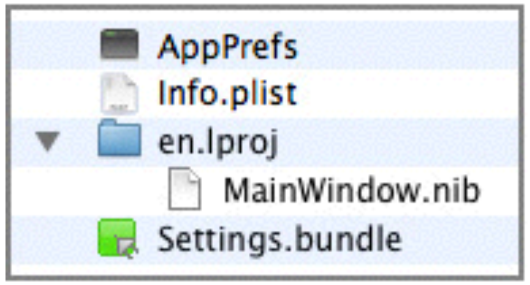
- *AppName*.app contains app itself and resources. Readonly access.
- Documents stores user generated content.
- Library contains non user data.
- Temp used to store temporary files that app doesn't need to persist between launches of the app.



Bundle directory

- can contain executable code,resources, frameworks and libraries, plug-ins, other bundles and also Info.plist
- Resources are placing in **Resources** subdir
- Localised files are put into subdirectories of Resources that have extension of lproj and name corresponding to a **language/locale**
- [NSBundle mainBundle]

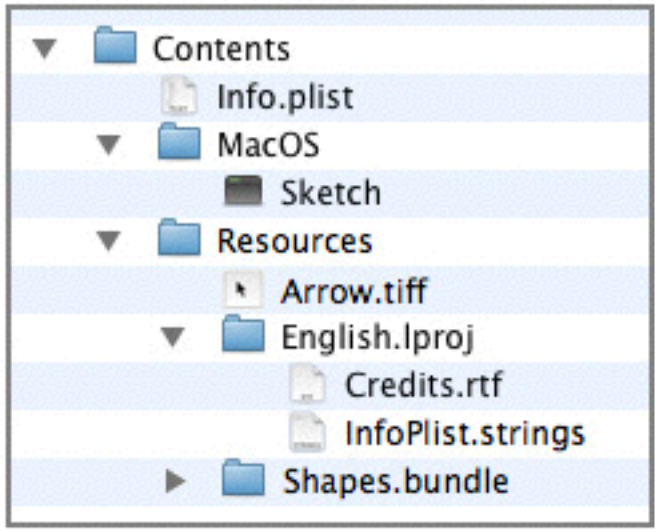
AppPrefs.app



- AppPrefs
- Info.plist
- ▼ en.lproj
  - MainWindow.nib
- Settings.bundle

iPhone OS

Sketch.app



- ▼ Contents
  - Info.plist
  - ▼ MacOS
    - Sketch
  - ▼ Resources
    - Arrow.tiff
    - ▼ English.lproj
      - Credits.rtf
      - InfoPlist.strings
    - ▶ Shapes.bundle

Mac OS X

Documents directory

- Store files you might want the user to create, import, delete or edit.
- Is accessible to user so don’t store files you don’t want to expose
- Inbox subdirectory stores documents that was asked to open by other apps. Can read and delete, but can’t add new or modify existing.
- Contents are backed by iTunes

Library directory

- You use this directory to store data files, caches, resources, preferences, and even user data in some specific situations.
- Subdir named Application Support used to store all data files except user's documents. Examples: templates,configs, data files, modified versions of resources from main bundle, downloadable content like game levels.
- Caches subdir is used to store some files that you can recreate easily. Not backed by iTunes and can be wiped anytime accept while app is running.
- Preferences contains app-specific preference files. Don't create files here yourself. Instead use NSUserDefaults
- Contents are backed by iTunes

Security: how to protect your files

- Sandbox already does most of the job
- Access to files is granted by mix of permissions and access control lists
- Files can be encrypted on disk
- Jailbroken devices can’t be considered as safe

NSFileManager

- use it to locate, create, copy, and move files and directories
- use either NSURL or NSString objects when specifying the location of files
- can be called from multiple threads safely
- accessible through singleton

NSURL

```
NSURL *baseURL = [NSURL URLWithString:@“file://path/to/some_folder/"];
NSURL *url = [NSURL URLWithString:@"folder/file.txt" relativeToURL:baseURL];

NSLog(@"url = %@", url);

NSURL *fileURL = [NSURL fileURLWithPath:@“path/to/some_folder/folder/file.txt"];

NSLog(@"fileURL = %@", fileURL);
```

Console output

```
2015-10-28 12:08:34.034 TutorialApplication[4132:184584] file://path/to/some_folder/folder/file.txt
2015-10-28 12:08:34.034 TutorialApplication[4132:184584] file://path/to/some_folder/folder/file.txt
```

Determining If A File Exists

```
    NSFileManager *fileManager = [NSFileManager defaultManager];

    NSString *documentsPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) firstObject];
    NSString *filePath = [documentsPath stringByAppendingPathComponent:@“file.txt”];

    BOOL fileExists = [fileManager fileExistsAtPath:filePath];

    NSLog(@"%d", fileExists);
```

Console output

```
2015-10-28 12:08:34.034 TutorialApplication[4132:184584] 0
```

Listing All Files in a Directory

```
NSFileManager *fileManager = [NSFileManager defaultManager];

NSURL *bundleURL = [[NSBundle mainBundle] bundleURL];

NSError *error = nil;
NSArray *contents = [fileManager contentsOfDirectoryAtURL:bundleURL
                               includingPropertiesForKeys:@[]
                                                  options:NSDirectoryEnumerationSkipsHiddenFiles
                                                    error:&error];

if ( error ) {
    NSLog(@"%s NSFileManager error: %@", __PRETTY_FUNCTION__, error);
}

NSLog(@"%@", contents);
```

Console output

```
2015-10-28 12:15:48.683 TutorialApplication[4195:198231] (
    "file:///Users/wirrwarr/Library/Developer/CoreSimulator/Devices/C322A129-B5FD-41DF-9487-A6097C5EDAF2/data/
Containers/Bundle/Application/3E731D08-F02C-4AF7-809D-7D74279BD5AB/TutorialApplication.app/Base.lproj/",
    "file:///Users/wirrwarr/Library/Developer/CoreSimulator/Devices/C322A129-B5FD-41DF-9487-A6097C5EDAF2/data/
Containers/Bundle/Application/3E731D08-F02C-4AF7-809D-7D74279BD5AB/TutorialApplication.app/Info.plist",
    "file:///Users/wirrwarr/Library/Developer/CoreSimulator/Devices/C322A129-B5FD-41DF-9487-A6097C5EDAF2/data/
Containers/Bundle/Application/3E731D08-F02C-4AF7-809D-7D74279BD5AB/TutorialApplication.app/PkgInfo",
    "file:///Users/wirrwarr/Library/Developer/CoreSimulator/Devices/C322A129-B5FD-41DF-9487-A6097C5EDAF2/data/
Containers/Bundle/Application/3E731D08-F02C-4AF7-809D-7D74279BD5AB/TutorialApplication.app/TutorialApplication"
)
```

Creating a Directory

```
    NSFileManager *fileManager = [NSFileManager defaultManager];
    NSString *documentsPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, 
NSUserDomainMask, YES) firstObject];
    NSString *logsPath = [documentsPath stringByAppendingPathComponent:@"logs"];
    if (![fileManager fileExistsAtPath:logsPath]) {
        NSError *error = nil;
        [fileManager createDirectoryAtPath:logsPath withIntermediateDirectories:NO attributes:nil 
error:&error];
        
        if ( error ) {
            NSLog(@"%s NSFileManager error: %@", __PRETTY_FUNCTION__, error);
        }
    }
    
    NSLog(@"%d", [fileManager fileExistsAtPath:imagesPath]);
```

Console output

```
2015-10-28 12:47:49.636 TutorialApplication[4265:230718] 1
```

Deleting a File

```
    NSFileManager *fileManager = [NSFileManager defaultManager];
    NSString *documentsPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) firstObject];
    NSString *filePath = [documentsPath stringByAppendingPathComponent:@"image.png"];
    NSError *error = nil;

    if (![fileManager removeItemAtPath:filePath error:&error]) {
        NSLog(@"[Error] %@ (%@)", error, filePath);
    }
```

Console output

```
2015-10-28 12:49:04.304 TutorialApplication[4278:232264] [Error] Error Domain=NSCocoaErrorDomain Code=4 "“image.png” couldn’t be
removed." UserInfo={NSFilePath=/Users/wirrwarr/Library/Developer/CoreSimulator/Devices/C322A129-B5FD-41DF-9487-A6097C5EDAF2/data/
Containers/Data/Application/3EED714B-EA2E-44DD-834B-894506C07DAE/Documents/image.png, NSUserStringVariant=(
    Remove
), NSUnderlyingError=0x7fee6151faf0 {Error Domain=NSPOSIXErrorDomain Code=2 "No such file or directory"}} (/Users/wirrwarr/
Library/Developer/CoreSimulator/Devices/C322A129-B5FD-41DF-9487-A6097C5EDAF2/data/Containers/Data/Application/3EED714B-
EA2E-44DD-834B-894506C07DAE/Documents/image.png)
```

Creating an Empty File

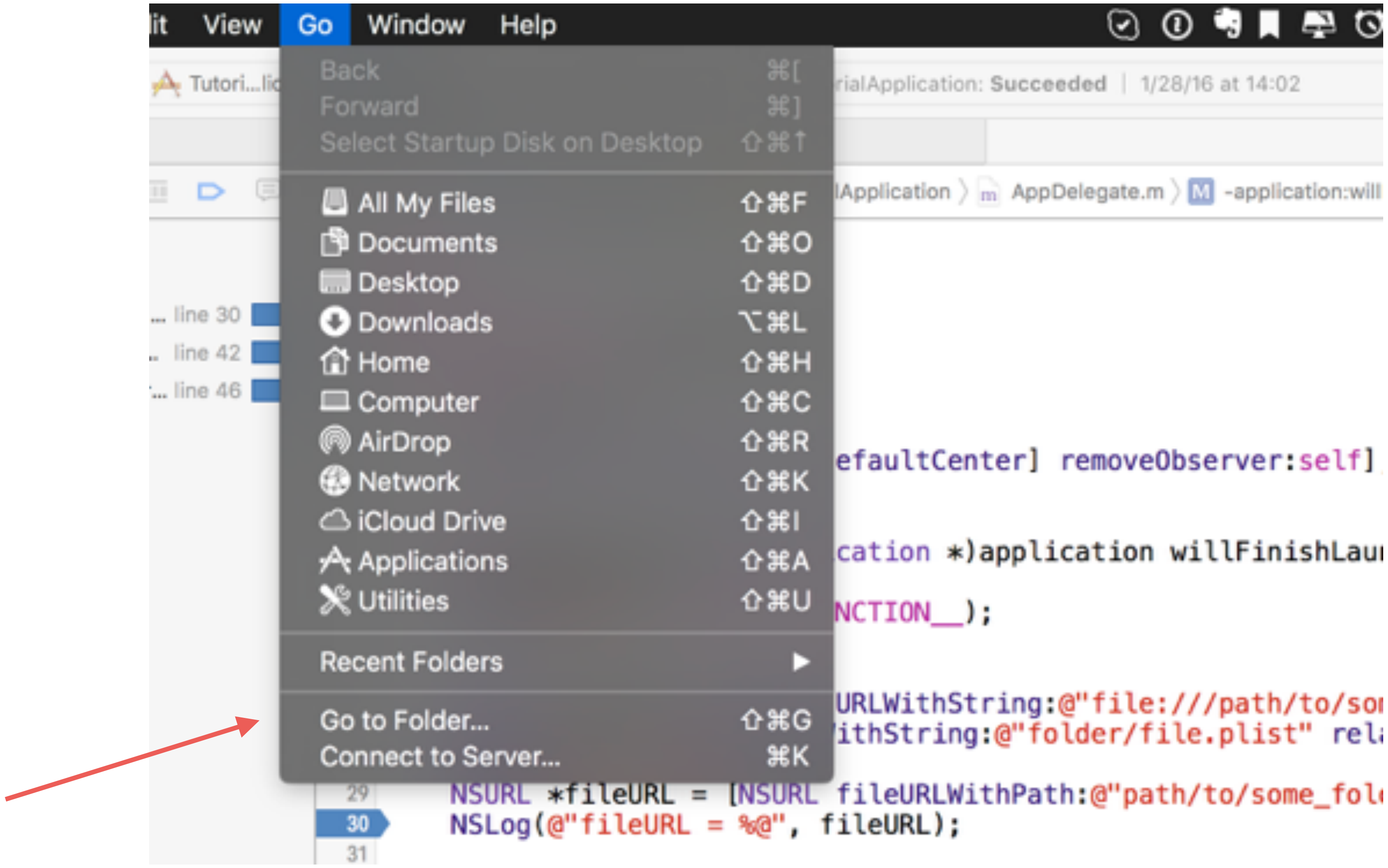
```
    NSFileManager *fileManager = [NSFileManager defaultManager];
    NSString *documentsPath = [NSSearchPathForDirectoriesInDomains(NSDocumentDirectory,
NSUserDomainMask, YES) firstObject];
    NSString *logsFolderPath = [documentsPath stringByAppendingPathComponent:@"logs"];
    if (![fileManager fileExistsAtPath:logsFolderPath]) {
        NSError *error = nil;
        [fileManager createDirectoryAtPath:logsFolderPath withIntermediateDirectories:NO
attributes:nil error:&error];

        if ( error ) {
            NSLog(@"%s NSFileManager error: %@", __PRETTY_FUNCTION__, error);
        }
    }

    NSString *logsFilePath = [documentsPath stringByAppendingPathComponent:@"logs.txt"];
    [fileManager createFileAtPath:logsFilePath contents:[NSData data] attributes:nil];

    BOOL isDirectory;
    NSLog(@"File exists: %d isDirectory:%d", [fileManager fileExistsAtPath:logsFilePath
isDirectory:&isDirectory], isDirectory);
```

Console output

```
2015-10-28 13:36:35.806 TutorialApplication[4617:273976] File exists: 1 isDirectory:0
```

Accessing iOS simulator directory

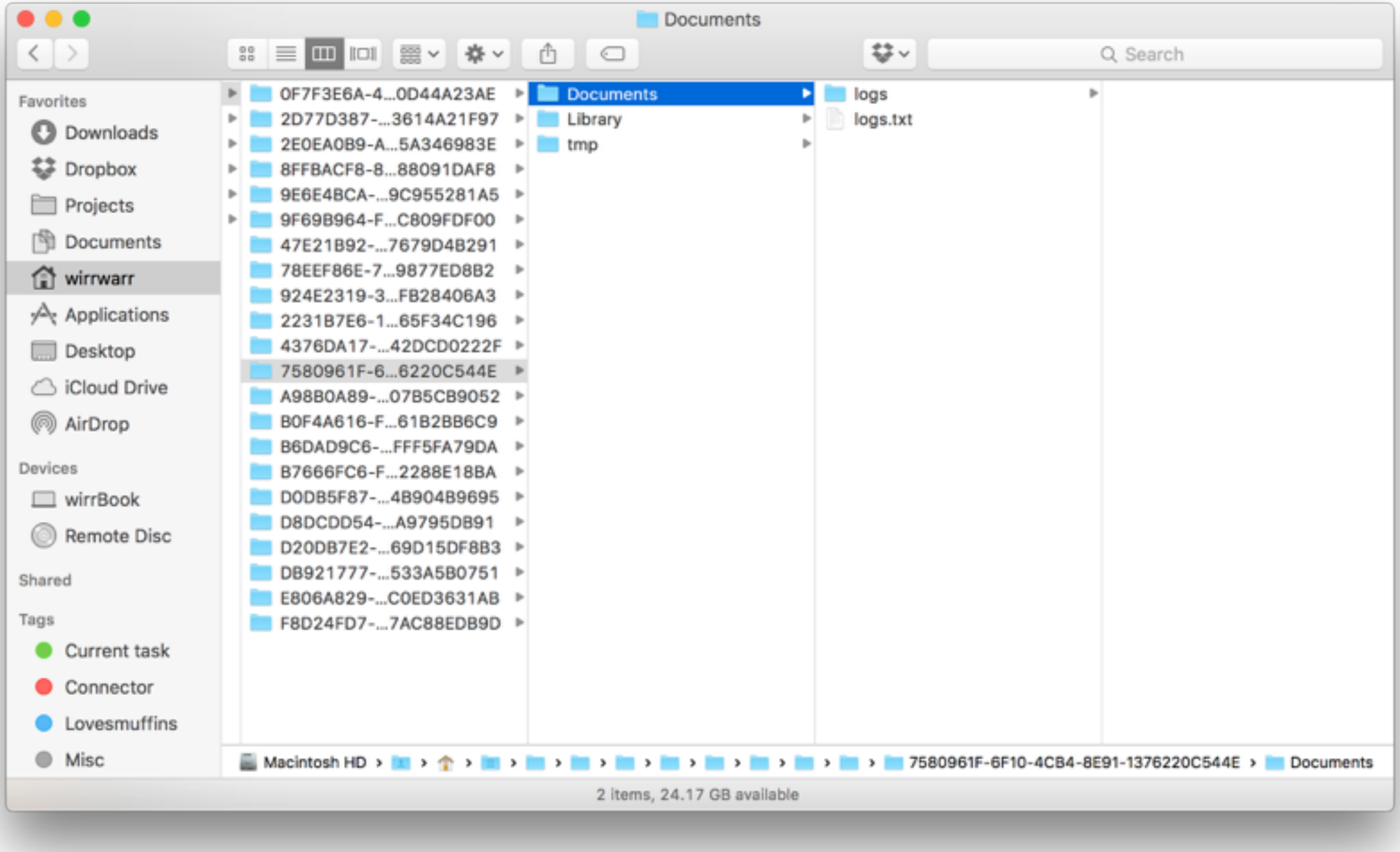
~/Library/Developer/CoreSimulator/Devices/{*Device_Identifier*}/data/Containers/Data/Application/{*App_Identifier*}/Documents/MyApp.sqlite



rialApplication: Succeeded | 1/28/16 at 14:02

```
efaultCenter] removeObserver:self]

cation *)application willFinishLau

NCTION__);

URLWithString:@"file:///path/to/so
ithString:@"folder/file.plist" rel
NSURL *fileURL = [NSURL fileURLWithPath:@"path/to/some_fol
NSLog(@"fileURL = %@", fileURL);
```

Location of logs.txt



0F7F3E6A-4...0D44A23AE
2D77D387-...3614A21F97
2E0EA0B9-A...5A346983E
8FFBACF8-8...88091DAF8
9E6E4BCA-...9C955281A5
9F69B964-F...C809FDF00
47E21B92-...7679D4B291
78EEF86E-7...9877ED8B2
924E2319-3...FB28406A3
2231B7E6-1...65F34C196
4376DA17-...42DCD0222F
7580961F-6...6220C544E
A98B0A89-...07B5CB9052
B0F4A616-F...61B2BB6C9
B6DAD9C6-...FFF5FA79DA
B7666FC6-F...2288E18BA
D0DB5F87-...4B904B9695
D8DCDD54-...A9795DB91
D20DB7E2-...69D15DF8B3
DB921777-...533A5B0751
E806A829-...C0ED3631AB
F8D24FD7-...7AC88EDB9D

Documents
Library
tmp

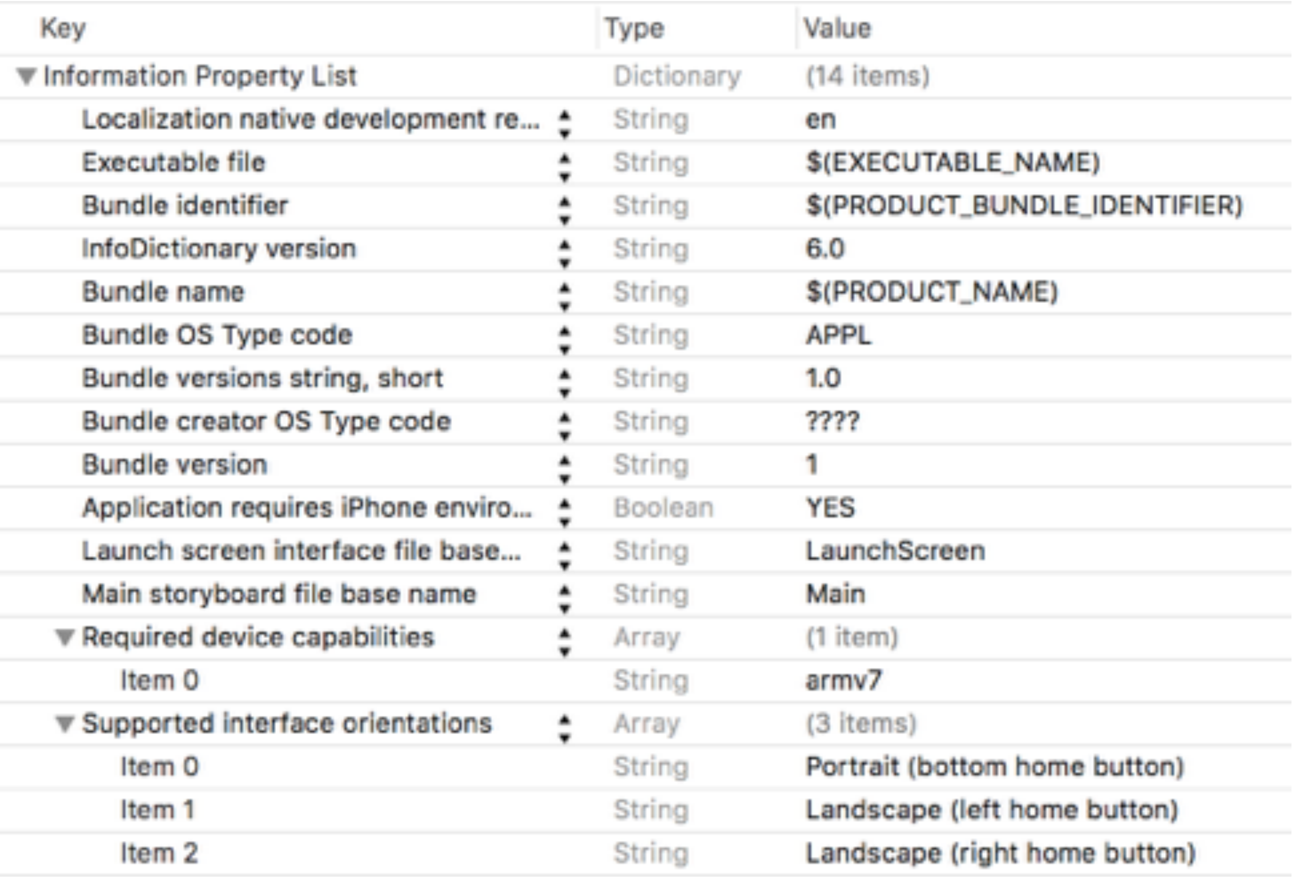
logs
logs.txt

Macintosh HD › › › › › › › › › › › › 7580961F-6F10-4CB4-8E91-1376220C544E › Documents

2 items, 24.17 GB available

What is .plist file?

- Property list
- XML structured
- Can store arrays and dictionaries with values Boolean, data, date, number, string
- Typically stores information about app and its configuration
- **Don’t use as a persistent store!**



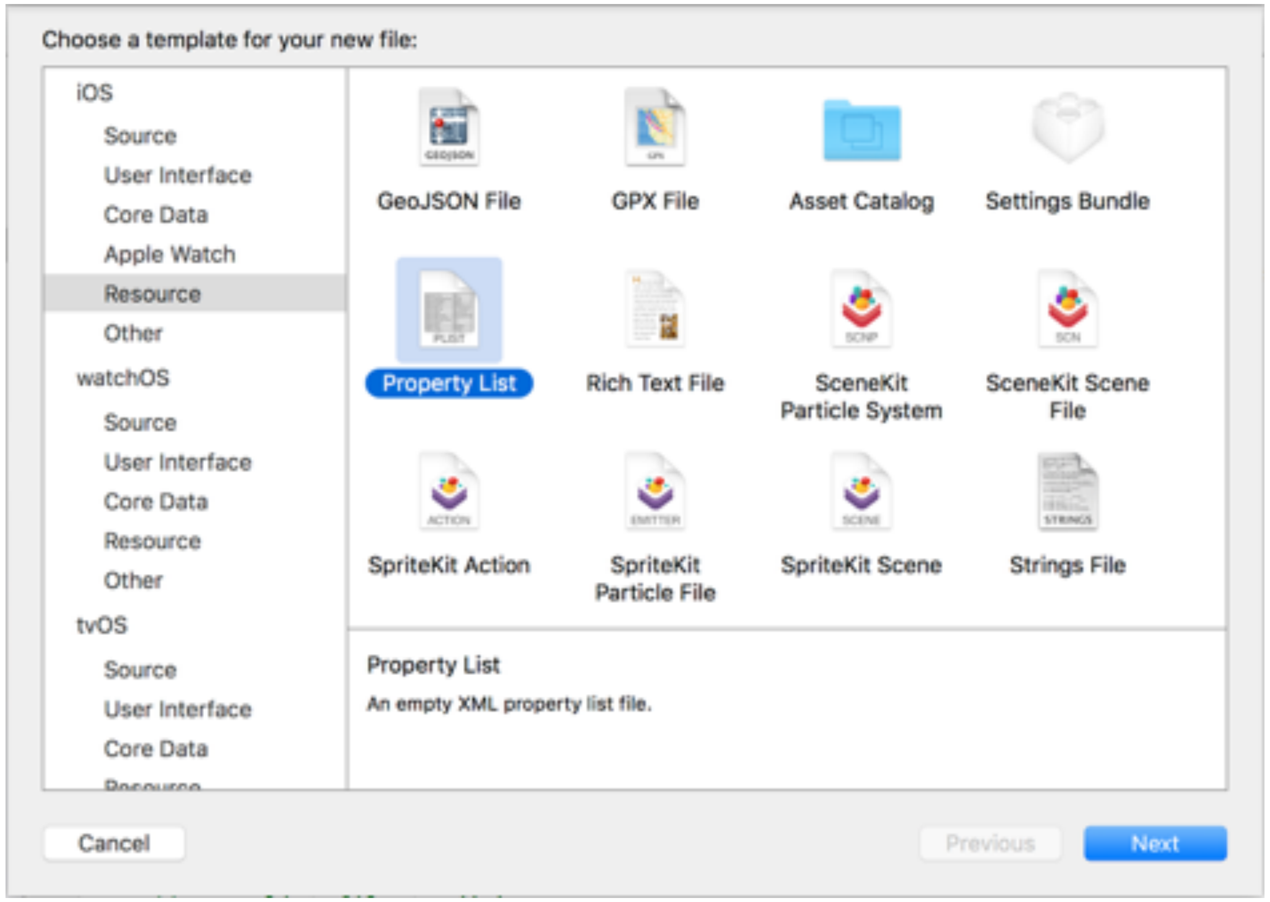
Key	Type	Value
▼ Information Property List	Dictionary	(14 items)
Localization native development re...	String	en
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
▼ Required device capabilities	Array	(1 item)
Item 0	String	armv7
▼ Supported interface orientations	Array	(3 items)
Item 0	String	Portrait (bottom home button)
Item 1	String	Landscape (left home button)
Item 2	String	Landscape (right home button)

Application’s Info.plist file

Reading and writing .plist files

```
//read plist from disk
NSDictionary *plistDictionary = [NSDictionary dictionaryWithContentsOfFile:logsFilePath];

//save plist file to disk
BOOL writtenSuccessfully = [plistDictionary writeToFile:logsPath atomically:YES];
```

- you can create a plist from XCode menu File->New->File-> Resource



Property List

An empty XML property list file.

Homework

Create an empty plist file from Xcode menu named logs.plist.

In runtime copy this file into the app Documents directory to be able to edit it

Create a PMRParty class with appropriate fields to represent party in our app.

Save parties into the plist.

Thank you for attention!

mail: mkramskoy@gmail.com
skype: wirrwarr74

Useful links

- https://developer.apple.com/library/ios/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html#//apple_ref/doc/uid/TP40010672-CH2-SW18
- http://nshipster.com/nsfilemanager/
- https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSFileManager_Class/
- http://nshipster.com/nscoding/