**iOS File Manager on top of NSFileManager for simplifying files management.** It provides many static methods for executing most common operations with few lines of code. It works by default in the Documents directory to allow use of relative paths, but it's possible to work easily on any other directory.

**Installation**

**CocoaPods:**

pod 'FCFileManager'

**Manual install:**

Copy FCFileManager.h and FCFileManager.m to your project.

**Features**

* **Build paths** relative to absolute directories *(FCFileManager works by default in the Documents directory, so you must build absolute paths only if you need to work outside of the Documents directory)*
* **Copy** files/directories
* **Create** files/directories
* **Check if** files/directory **exists**
* **Get** files/directories **attributes** *(creation date, size, ...)*
* **List** files/directories
* **Move** files/directories
* **Read/Write** files content in different formats *(arrays, custom models, data, dictionaries, images, json, strings, ... )*
* **Read/Write xattr** *(Extended File Attributes)*
* **Read** images metadata, **EXIF** data, **TIFF** data
* **Remove** files/directories
* **Rename** files/directories
* Directories are created on the fly
* **Error handling** as using NSFileManager

See [FCFileManager.h](https://github.com/fabiocaccamo/FCFileManager/blob/master/FCFileManager/FCFileManager.h) for all of the methods.

**Usage examples**

**Build path:**

//my file path, this will be automatically used as it's relative to the Documents directory

NSString \*testPath = @"test.txt";

//my file path relative to the temporary directory path

NSString \*testPathTemp = [FCFileManager pathForTemporaryDirectoryWithPath:testPath];

/\*

All shortcuts suppported:

pathForApplicationSupportDirectory;

pathForCachesDirectory;

pathForDocumentsDirectory;

pathForLibraryDirectory;

pathForMainBundleDirectory;

pathForPlistNamed:(NSString \*)name; //look for {{ name }}.plist in the main bundle directory

pathForTemporaryDirectory;

\*/

**Copy file:**

//copy file from Documents directory (public) to ApplicationSupport directory (private)

NSString \*testPath = [FCFileManager pathForApplicationSupportDirectoryWithPath:@"test-copy.txt"];

[FCFileManager copyItemAtPath:@"test.txt" toPath:testPath];

**Create file:**

//create file and write content to it (if it doesn't exist)

[FCFileManager createFileAtPath:@"test.txt" withContent:@"File management has never been so easy!!!"];

**Create directories:**

//create directories tree for the given path (in this case in the Documents directory)

[FCFileManager createDirectoriesForPath:@"/a/b/c/d/"];

**Check if file exists:**

//check if file exist and returns YES or NO

BOOL testFileExists = [FCFileManager existsItemAtPath:@"test.txt"];

**Move file:**

//move file from a path to another and returns YES or NO

[FCFileManager moveItemAtPath:@"test.txt" toPath:@"tests/test.txt"];

**Read file:**

//read file from path and returns its content (NSString in this case)

NSString \*test = [FCFileManager readFileAtPath:@"test.txt"];

**Read/Write xattr (Extended File Attributes):**

//returns the string-value stored for the specified key, if the key doesn't exist returns nil

NSString \*value = [FCFileManager xattrOfItemAtPath:@"test.txt" getValueForKey:"uploaded"];

//set the specified string-value and returns a BOOL result of the operation

BOOL success = [FCFileManager xattrOfItemAtPath:@"test.txt" setValue:@"1" forKey:@"uploaded"];

**Read image EXIF data:**

//read image file from path and returns its EXIF data

NSDictionary \*exifData = [FCFileManager exifDataOfImageAtPath:@"test.jpg"];

**Remove file:**

//remove file at the specified path

[FCFileManager removeItemAtPath:@"test.txt"];

**Rename file:**

//rename file at the specified path with the new name

[FCFileManager renameItemAtPath:@"test.txt" withName:@"test-renamed.txt"];

**Write file:**

NSArray \*testContent = [NSArray arrayWithObjects:@"t", @"e", @"s", @"t", nil];

//write file at the specified path with content

[FCFileManager writeFileAtPath:@"test.txt" content:testContent];

**Get file/directory size:**

//get the file size in bytes

NSNumber \*fileSize = [FCFileManager sizeOfFileAtPath:@"test.txt"];

//get the directory size in bytes (including all subdirectories and files inside it)

NSNumber \*directorySize = [FCFileManager sizeOfDirectoryAtPath:@"/a/"];

**Get file/directory size formatted:**

//returns a human-readable file size formatted with the necessary suffix: bytes, KB, MB, GB, TB

NSString \*fileSizeFormatted = [FCFileManager sizeFormattedOfFileAtPath:@"test.txt"];