

This summarize how to handle a recorded video in AVFoundation.<sup>1</sup>

In `captureOutput:didFinishRecordingToOutputFileAtURL:fromConnections:error:`, we have a video in the argument of `didFinishRecordingToOutputFileAtURL:`. This is in a temporary path, so the video should be saved in “Photos”, which is assets in asset group called “Camera Roll”.<sup>2</sup>

A typical example<sup>3</sup> is to call `writeVideoAtPathToSavedPhotosAlbum:completionBlock:` of `ALAssetsLibrary` class, and in `completionBlock:`, to delete a video in a temporary folder.

```
- (void)
    captureOutput:(AVCaptureFileOutput *)captureOutput
    didFinishRecordingToOutputFileAtURL:(NSURL *)outputFileURL
    fromConnections:(NSArray *)connections
    error:(NSError *)error
{
    ...
    [
        [[ALAssetsLibrary alloc] init]
        writeVideoAtPathToSavedPhotosAlbum:outputFileURL
        completionBlock:^(NSURL *assetURL, NSError *error) {
            ...
            [[NSFileManager defaultManager] removeItemAtURL:outputFileURL error:nil];
            ...
        }
    ]
    ...
}
```

We like to extend the capability for handling a video:

1. Upload the video
2. Keep track of maintaining association between a video in “Photos” and the uploaded video

NetPut class is designed for Extension 1. The sqlite3 database is used for Extension 2. (See `DBManager` class.) A sample database called “video.db” is in a bundle called “db.bundle”. Add the bundle into Xcode and use it after copying into a “writable” folder.<sup>4</sup> Here is a typical example for the use of sqlite3.

---

<sup>1</sup> See “Starting a Recording”:

[https://developer.apple.com/library/ios/documentation/AudioVideo/Conceptual/AVFoundationPG/Articles/04\\_MediaCapture.html#//apple\\_ref/doc/uid/TP40010188-CH5-SW17](https://developer.apple.com/library/ios/documentation/AudioVideo/Conceptual/AVFoundationPG/Articles/04_MediaCapture.html#//apple_ref/doc/uid/TP40010188-CH5-SW17)

<sup>2</sup> To enumerate all assets in all asset groups, use `[OwnUtil lsAssetsLibrary]`.

<sup>3</sup> A good sample is here: <https://developer.apple.com/library/ios/samplecode/AVCam/Introduction/Intro.html>

<sup>4</sup> Document/ or Library/ folder is a good candidate. See

<https://developer.apple.com/library/ios/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html>

```
#import "DBManager.h"

...
@property (nonatomic) DBManger *dbManager;

...
_dbManager = [[DBManager alloc] initWithDatabaseFilename:@"video.db"];

...
BOOL b = [_dbManager
    insertVideoPath:videoPathInServer
    withNameInCameraRoll:filenameInCameraRoll
    withUserId:99
];
```

Of course, `videoPathInServer`, `filenameInCameraRoll`, and `99(UserId)` should be determined before a call to `insertVideoPath:withNameInCameraRoll:withUserId:`.<sup>5</sup> All are done in the typical example below, which extends [captureOutput:didFinishRecordingToOutputFileAtURL:fromConnections:error:.](#)

```
[[[ALAssetsLibrary alloc] init]
assetForURL:assetURL
resultBlock:^(ALAsset *asset) {
    //ALog(@"ALAsset: |%@", asset);
    ALAssetRepresentation *repr = [asset defaultRepresentation];
    //ALog(@"ALAssetRepresentation: |%@", repr);
    NSString *filenameInCameraRoll = repr == nil ? nil : repr.filename;
    ALog(@"INFO: filename in Camera Roll: |%@", filenameInCameraRoll);
    if (!filenameInCameraRoll) {
        ALog(@"ERR: filenameInCameraRoll is nil");
        return;
    }

    dispatch_async(dispatch_get_main_queue(), ^{
        NSURL *url = [NSURL URLWithString:@"http://ownphones.com/unity/net_put.php"];
        NetPut *netPut = [[NetPut alloc] init];
        BOOL b = [netPut
            send:outputFileURL
            to:url
            headers:nil
            onProgressUpdate:^(int progress) {
                ALog(@"progress = %d %%", progress);
            }
            onEnd:^(NSInteger statusCode, NSString *remoteFilePath) {
                // delete a video in outputFileURL
                NSError *err = nil;
```

---

<sup>5</sup> A method `getDbURL` gives a URL for database in a writable folder.

```

        [[NSFileManager defaultManager]
         removeItemAtURL:outputFileURL
         error:&err
    ];
    if (err)
        ALog(@"ERR: |%@", err);

    if (statusCode != 200) {
        ALog(@"ERR: Status Code = %@", OPInteger(statusCode));
        return;
    }
    NSString *videoPathInServer = remoteFilePath;
    ALog(@"Video Path in Server: |%@", videoPathInServer);
    BOOL b = [self.dbManager
              insertVideoPath:videoPathInServer
              withNameInCameraRoll:filenameInCameraRoll
              withUserId:99
    ];
    ALog(@"INFO: Inserted? %s", b ? "Okay" : "Oops");
    if (b) {
        // backup db
        NSURL *dbURL = [self.dbManager getDbURL];
        [[NetPut alloc] init]
            send:dbURL
            to:url
            headers:nil
            onProgressUpdate:nil
            onEnd:^(NSInteger statusCode, NSString* remoteFilePath) {
                if (statusCode != 200) {
                    ALog(@"ERR: Status Code = %@", OPInteger(statusCode));
                    return;
                }
                ALog(@"INFO: db(|%@", backup done as |%@",
                    dbURL, remoteFilePath
                );
            }
    ];
    }
    ];
    if (!b) {
        ALog(@"ERR: NetPut:send(|%@", to(|%@",...: failed", outputFileURL, url);
    }
    });
}

failureBlock:^(NSError *err) {
    ALog(@"ALAssetsLibrary assetURL:resultBlock:failureBlock: |%@", err);
}
];

```

## Notes<sup>6</sup>

1. To determine `filenameInCameraRoll`, use `assetForURL:resultBlock:failureBlock:` in `ALAssetsLibrary` class.
2. `NetPut` class is used to upload a video; internally `NSURLConnection`.<sup>7</sup>
3. `NSURLConnection` should be called in a main thread, which is why `dispatch_async(dispatch_get_main_queue(), ^{...})` is called.
4. In `onProgressUpdate`, one can update progress in uploading.
5. Optionally, updated database can be uploaded too with another call in `NetPut` class.

---

<sup>6</sup> All are packed in `AVCam2.zip` which is an extension of `AVCam.zip` (See a footnote 3.)

<sup>7</sup> `NetPut` class is connected with a PHP script in a server: `net_put.php`, `dev_filter.php`.