Necessary files

For Xcode:

```
OwnXcode/
    OwnUtil.h
    OwnUtil.m
    NetPut.h
    NetPut.m
```

For PHP:

```
OwnXcode/php_scripts/
    net_put.php
    dev_filter.phpi
```

Usage

1. Add **OwnXcode** directory into Xcode
2. In a ".m" file which uploads a (video) file,

```
#include "NetPut.h"
……
// upload a video with videoFileURL
// set serverURL
NSURL *url = [NSURL URLWithString:@"http://ownphones.com/unity/net_put.php"];
// upload via NetPut put:to:onEnd: call
[[[NetPut alloc] init]
    put:videoFileURL
    to:url
    onEnd:^(NSURL *videoFileURL) {
        // after the end of upload
    }
];
```

where onEnd is a block (in objective-c), which is called at the end of upload, so that is the place for clean-up, for example,

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

Note that

1. Two PHP files should be approriate place; for example, http://ownphones.com/unity/.
2. NetPut can upload multiple files at the same time; see **Multiple Files** below.
3. On successful upload, "Debug Area" displays the string:
   "OwnPhones: NetPut: Okay" (23 bytes)
   For more, see **Details on net_put.php** below.

Here is a typical example.

```
#include “NetPut.h”
……
```
// upload a video with videoFileURL
// set serverURL
```
NSURL *url = [NSURL URLWithString:@"http://ownphones.com/unity/net_put.php"];
```
// upload via **`NetPut put:to:onEnd:`** call
```
[[[NetPut alloc] init]
	put:videoFileURL
	to:url
	onEnd:^(NSURL *videoFileURL) {
		// after the end of upload
		NSError *err;
		[[NSFileManager defaultManager]
			removeItemAtURL:videoFileURL
			error:&err
		];
		if (err)
			NSLog(@"ERR: |%@|", err);
	}
];
```

Details on net_put.php
It only accepts PUT method only for request, and directly copy upload stream to a file, which is contrary to POST method.

There are two constants:
```
	define('CONTENT_LENGTH_LIMIT', 10000000);
	define('VIDEO_PATH', '../video/');  // sys_get_temp_dir()
```
“../video/” is a relative path for video files.

The video file’s name looks “XXX556f66c866f4e” with a prefix “XXX”, so the name itself can be changed in net_put.php; search for
```
	$video_pathname = VIDEO_PATH . '/XXX' . uniqid();
```

Known Issues
1. No upload if the App is terminated manually or by a system.
2. Determine if the upload is valid or not only after all data are accepted.

Multiple Files

NetPut can upload multiple files at the same time; each upload will be done in the background thread. So there should be warning on naming on the video file.

Typically, the video file is created in a temporary directory as

```
NSString *videoFilePath = [
        NSTemporaryDirectory()
        stringByAppendingPathComponent:[
                @"movie"
                stringByAppendingPathExtension:@"mov"
        ]
];
```

which is an excerpt from AVCamViewController.m in AVCam.zip (see **References** below.)

For multiple files, the name of each file should be set separately as “movie0000.mov”, “movie0001.mov”, …, etc. So the above should be changed as

```
NSString *videoFilePath = [
        NSTemporaryDirectory()
        stringByAppendingPathComponent:[
                [NSString stringWithFormat:@"movie%04d", self.nMovie++]
                stringByAppendingPathExtension:@"mov"
        ]
];
```

, where a property `nMovie` is introduced and is set to 0 initially.

```
@property (nonatomic) int nMovie;
…
self.nMovide = 0;
```

References

For recording a video (AVCam.zip)

https://developer.apple.com/library/ios/samplecode/AVCam/Introduction/Intro.html

For URL connection (SimpleURLConnections.zip)

https://developer.apple.com/library/ios/samplecode/SimpleURLConnections/Introduction/Intro.html

For task in a background via `dispatch_async` with block, see the above AVCam.zip as an example, and, for details, the document on Objective-C by Apple, “Working with Blocks (pp. 104-116), “Schedule Blocks on Dispatch Queues with Grand Central Dispatch (p. 116)

https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/ProgrammingWithObjectiveC.pdf