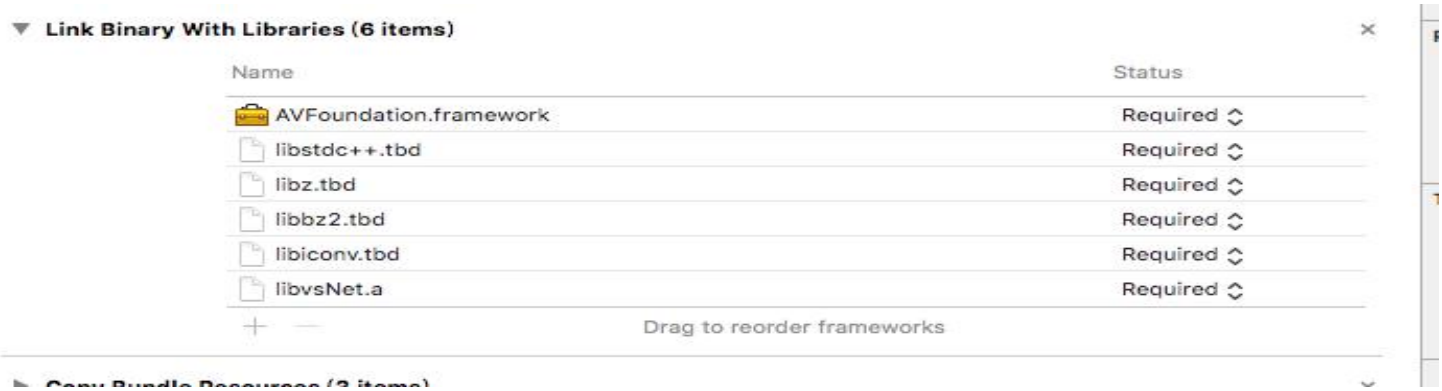
IOS VSNet Library instructions

1. Initialization library

1）XCODE: Enable Bitcode yes 改为 no

2）Dependency library



3）Library Initialization

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    [[VSNet shareinstance] PPPP_Initialize];
    [[VSNet shareinstance] XQP2P_NetworkDetect];
    [[VSNet shareinstance] XQP2P_Initialize];
    return YES;
}
```

2. Device management

1）Connecting to device

```
int nRet = [[VSNet shareinstance] start:strDID  withUser:@"admin" withPassWord:strPWD initializeStr:nil LanSearch:1];
if (nRet == 0) {
    //连接不成功，3秒后再试一次
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(3 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{
       [[VSNet shareinstance] start:strDID  withUser:@"admin" withPassWord:strPWD initializeStr:nil LanSearch:1];
       [[VSNet shareinstance] setStatusDelegate:strDID withDelegate:self];//设置代理接收设备状态
       [[VSNet shareinstance] setControlDelegate:strDID withDelegate:self];//设置代理接收所发指令设备回复
    });

}
else{
    [[VSNet shareinstance] setStatusDelegate:strDID withDelegate:self];
    [[VSNet shareinstance] setControlDelegate:strDID withDelegate:self];
}
```

2）Disconnect

```
[[VSNet shareinstance] stop:strDID];
```

3）Connection　status

```
#pragma mark  VSNetStatueProtocol
- (void) VSNetStatus: (NSString*) deviceIdentity statusType:(NSInteger) statusType status:(NSInteger) status
{
    NSLog(@"PPPPStatus ..... strDID: %@, statusType: %ld, status: %ld", deviceIdentity, statusType, status);
    if (statusType == MSG_NOTIFY_TYPE_PPPP_STATUS) {
        //如果是ID号无效，则停止该设备的P2P
        if (status == PPPP_STATUS_INVALID_ID
            || status == PPPP_STATUS_CONNECT_TIMEOUT
            || status == PPPP_STATUS_DEVICE_NOT_ON_LINE
            || status == PPPP_STATUS_CONNECT_FAILED
            || status == PPPP_STATUS_INVALID_USER_PWD)
        {
            NSLog(@"设备连接失败");
        }
        else if(PPPP_STATUS_ON_LINE == status){
            NSLog(@"设备在线");
        }
        else if(PPPP_STATUS_CONNECTING == status){
            NSLog(@"连接中...");
        }
        else if(PPPP_STATUS_INITIALING == status){
            NSLog(@"正在初化");
        }
        return;
    }
}
```

4）Device password management

（4.1）Reset device password

```
NSString *cmdStr = [NSString stringWithFormat:@"set_users.cgi?&user1=%@&user2=%@&user3=%@&pwd1=%@&pwd2=%@&pwd3=%@&", @"", @"",
    @"admin", @"", @"", m_strPwd];
[[VSNet shareinstance] sendCgiCommand:cmdStr withIdentity:m_strDID];
```

（4.2）Reset device password

```
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"UserPwdSetViewController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if (comType == CGI_IESET_USER && [deviceIdentity isEqualToString:m_strDID]) {
        NSInteger result = [[APICommon stringAnalysisWithFormatStr:@"result=" AndRetString:retString] integerValue];
        if (result == 0){
            [[VSNet shareinstance] sendCgiCommand:@"reboot.cgi?" withIdentity:m_strDID];
            [self EditP2PCameraInfo:NO Name:self.cameraName DID:self.m_strDID User:@"admin" Pwd:self.m_strPwd OldDID:self.m_strDID];
        }
        else{
            NSLog(@"修改密码失败");
        }
    }
```

5）Device WiFi managment

（5.1）Get the current device WIFI

```
[[VSNet shareinstance] sendCgiCommand:@"get_params.cgi?"withIdentity:self.m_strDID];
```

(5.2) return　Get the current device WIFI

```
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"WifiSettingViewController: VSNet返回数据 UID:%@,comType:%ld",deviceIdentity,(long)comType); 2 ⚠ Data argument not used by format stri...
    NSString *string = [[NSString alloc] initWithCString:buffer encoding:NSUTF8StringEncoding];
    if ([deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM)
    {
        NSInteger result = [[NSString subValueByKeyString:@"result=" fromRetString:string] integerValue];
        if (result != 0) {
            NSLog(@"数据异常!");
            return;
        }

        m_strSSID = [NSString subValueByKeyString:@"wifi_ssid=" fromRetString:string];
        m_channel = [[NSString subValueByKeyString:@"wifi_channel=" fromRetString:string] intValue];
        m_authtype = [[NSString subValueByKeyString:@"wifi_authtype=" fromRetString:string] intValue];
        m_strWEPKey = [NSString subValueByKeyString:@"wifi_key1=" fromRetString:string];
        m_strWPA_PSK = [NSString subValueByKeyString:@"wifi_wpa_psk=" fromRetString:string];
    }
```

（5.3）Get device WIFI list

```
[[VSNet shareinstance] sendCgiCommand:@"wifi_scan.cgi?"
withIdentity:self.m_strDID];
[[VSNet shareinstance] setControlDelegate:self.m_strDID withDelegate:self];
```

(5.4) return　Get device WIFI list

```
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"WifiSettingViewController: VSNet返回数据 UID:%@,comType:%ld",deviceIdentity,(long)comType); 2 ⚠ Data argument not used by for
    NSString *string = [[NSString alloc] initWithCString:buffer encoding:NSUTF8StringEncoding];
    if ([deviceIdentity isEqualToString:m_strDID] && comType == CGI_IESET_WIFISCAN) {
        if (string == nil) {
            if (retString != nil) {
                string = retString;
            } else {
                string = [NSString stringWithFormat:@"%s",buffer];
            }
        }
        NSLog(@"无线wifi返回数据: \nUID = %@,类型 = %ld,buff = %@",deviceIdentity,(long)comType,string);
        NSInteger result = [[NSString subValueByKeyString:@"result=" fromRetString:string] integerValue];
        if (result != 0) {
            NSLog(@"数据异常!");
            return;
        }
```

（5.5）Set device WIFI

```
NSString *cmd = [NSString stringWithFormat:@"set_wifi.cgi?
    enable=1&ssid=%@&encrypt=0&defkey=0&key1=%s&key2=&key3=&key4=&authtype=%d&keyformat=0&key1_bits=0&key2_bits=0&key3_bits=0&key4_bi
    s=0&channel=%d&mode=0&wpa_psk=%s&",m_strSSID,pkey,m_security,m_channel,pwpa_psk];

NSString *sendSSID = [cmd stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];

[[VSNet shareinstance] sendCgiCommand:sendSSID withIdentity:m_strDID];
[[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
```

6）Device firmware upgrade

```
        NSLog(@"%@=+++%@",self.firmware_server,self.firmware_file);
        NSString *cmd = [NSString stringWithFormat:@"auto_download_file.cgi?
            server=%@&file=%@&type=%d&resevered1=&resevered2=&resevered3=&resevered4=&",self.firmware_server,self.firmware_file,0];
        [[VSNet shareinstance] sendCgiCommand:cmd withIdentity:self.str_uid];
```

7）Restart the device

```
[[VSNet shareinstance] sendCgiCommand:@"reboot.cgi?" withIdentity:strUID];
```

8）Equipment parameters

（8.1）Get device parameters

```
90
91      [[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
92      [[VSNet shareinstance] sendCgiCommand:@"get_params.cgi?" withIdentity:m_strDID];
```

（8.2）return　Get device parameter

```
#pragma mark - VSNetControlProtocol
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"DateTimeController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if ( [deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM) {
        m_timeZone = -[[APICommon stringAnalysisWithFormatStr:@"tz=" AndRetString:retString] integerValue];
        m_dateTime = [[APICommon stringAnalysisWithFormatStr:@"now=" AndRetString:retString] integerValue];
        m_Timing = [[APICommon stringAnalysisWithFormatStr:@"ntp_enable=" AndRetString:retString] integerValue];
        m_timingSever = [APICommon stringAnalysisWithFormatStr:@"ntp_svr=" AndRetString:retString];
```

9）Equipment alarm

（9.1）Get alarm parameters

```
[[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
[[VSNet shareinstance] sendCgiCommand:@"get_params.cgi?" withIdentity:m_strDID];
```

（9.2）Return　get the alarm parameters

```
#pragma mark - VSNetControlProtocol
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"AlarmController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if ( [deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM)
    {
        m_motion_armed = [[NSString subValueByKeyString:@"alarm_motion_armed=" fromRetString:retString] intValue];
        m_motion_sensitivity = [[NSString subValueByKeyString:@"alarm_motion_sensitivity=" fromRetString:retString] intValue];
        m_input_armed = [[NSString subValueByKeyString:@"alarm_input_armed=" fromRetString:retString] intValue];
        m_ioin_level = [[NSString subValueByKeyString:@"alarm_ioin_level=" fromRetString:retString] intValue];
        m_alarmpresetsit= [[NSString subValueByKeyString:@"alarm_presetsit=" fromRetString:retString] intValue];
        m_iolinkage= [[NSString subValueByKeyString:@"alarm_iolinkage=" fromRetString:retString] intValue];
        m_ioout_level= [[NSString subValueByKeyString:@"alarm_ioout_level=" fromRetString:retString] intValue];
        m_mail = [[NSString subValueByKeyString:@"alarm_mail=" fromRetString:retString] intValue];
        m_snapshot = [[NSString subValueByKeyString:@"alarm_snapshot=" fromRetString:retString] intValue];
        m_upload_interval = [[NSString subValueByKeyString:@"alarm_upload_interval=" fromRetString:retString] intValue];
        m_record = [[NSString subValueByKeyString:@"alarm_record=" fromRetString:retString] intValue];
        m_enable_alarm_audio = [[NSString subValueByKeyString:@"enable_alarm_audio=" fromRetString:retString] intValue];
        [self performSelectorOnMainThread:@selector(reloadTableView:) withObject:nil waitUntilDone:NO];
    }
}
```

（9.3）Set alarm parameters

```
NSString *cmd = [NSString stringWithFormat:@"set_alarm.cgi?
    enable_alarm_audio=%d&motion_armed=%d&motion_sensitivity=%d&input_armed=%d&ioin_level=%d&preset=%d&iolinkage=%d&ioout_level=%d&mai
    l=%d&record=%d&upload_interval=%d&schedule_enable=1&schedule_sun_0=-1&schedule_sun_1=-1&schedule_sun_2=-1&schedule_mon_0=-1&schedu
    le_mon_1=-1&schedule_mon_2=-1&schedule_tue_0=-1&schedule_tue_1=-1&schedule_tue_2=-1&schedule_wed_0=-1&schedule_wed_1=-1&schedule_w
    ed_2=-1&schedule_thu_0=-1&schedule_thu_1=-1&schedule_thu_2=-1&schedule_fri_0=-1&schedule_fri_1=-1&schedule_fri_2=-1&schedule_sat_0
    =-1&schedule_sat_1=-1&schedule_sat_2=-1&",
    0,m_motion_armed,m_motion_sensitivity,m_input_armed,m_ioin_level,m_alarmpresetsit,m_iolinkage,m_ioout_level,m_mail,
    1,m_upload_interval];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:self.m_strDID];
```

10）Device preset position

（10.1）Set device preset 0

```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET0];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

（10.2）Set device preset 1

```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET1];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

（10.3）Set device preset 2

```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET2];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

（10.4）Set device preset 3

```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_SET3];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

（10.5）Set device preset 4

```
NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_PREFAB_BIT_SET4,
    onestep];
[[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
```

（10.6）Call device preset

```
switch (((UIButton*)sender).tag) {
    case 100:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN0];
        break;
        case 101:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN1];
        break;
        case 102:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN2];
        break;
        case 103:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN3];
        break;
        case 104:
        cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=0&" ,CMD_PTZ_PREFAB_BIT_RUN4];
        break;
    default:
        break;
}

if (cgi) {
    [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:_strDID];
}
```

11）PTZ operation

（11.1）Cruise up and down

```
- (IBAction) btnUpDown:(id)sender
{
    if (m_bPtzIsUpDown) {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_UP_DOWN_STOP, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];

        btnUpDown.style = UIBarButtonItemStyleBordered;
        [_upDownBtn setImage:_arrowUpDownImg forState:UIControlStateNormal];
    }else {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_UP_DOWN, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];
        btnUpDown.style = UIBarButtonItemStyleDone;
        [_upDownBtn setImage:_arrowUpDownImgOn forState:UIControlStateNormal];
    }
```

（11.2）Cruise left and right

```
- (IBAction) btnLeftRight:(id)sender
{
    if (m_bPtzIsLeftRight) {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_LEFT_RIGHT_STOP, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];

        btnLeftRight.style = UIBarButtonItemStyleBordered;
        [_leftRightBtn setImage:_arrowLeftRightImg forState:UIControlStateNormal];
    }else {
        int onestep = 0;
        NSString *cgi = [NSString stringWithFormat:@"GET /decoder_control.cgi?command=%d&onestep=%d&" ,CMD_PTZ_LEFT_RIGHT, onestep];
        [[VSNet shareinstance] sendCgiCommand:cgi withIdentity:strDID];

        btnLeftRight.style = UIBarButtonItemStyleDone;
        [_leftRightBtn setImage:_arrowLeftRightImgOn forState:UIControlStateNormal];
    }
```

12）Image sensor parameter control

(12.1) Flip and mirror

```
NSString *cmd = [NSString stringWithFormat:@"camera_control.cgi?param=5&value=%d&",value];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:strDID];
```

(12.2) brightness

```
int f = sliderBrightness.value;
NSString *cmd = [NSString stringWithFormat:@"camera_control.cgi?param=1&value=%d&",f];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:strDID];
```

(12.3) Contrast

```
int f = sliderContrast.value;
NSString *cmd = [NSString stringWithFormat:@"camera_control.cgi?param=2&value=%d&",f];
[[VSNet shareinstance] sendCgiCommand:cmd withIdentity:strDID];
```

13）Device screenshot

（13.1）Get device screenshot

```
NSString *did = [cameraDic objectForKey:@STR_DID];
[[VSNet shareinstance] setControlDelegate:did withDelegate:self];
[[VSNet shareinstance] sendCgiCommand:@"snapshot.cgi?res=1&" withIdentity:did];
```

（13.2）return　Get device screenshot

```
# pragma mark VSNetControlProtocol
- (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
    charBuffer:(char *)buffer
{
    NSLog(@"CameraViewController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    switch (comType) {
        case CGI_IESET_SNAPSHOT:
        {
            NSData *image = [[NSData alloc] initWithBytes:buffer length:length];
            [self SnapshotCallback:image UID:deviceIdentity];
            break;
        }
        default:
            break;
    }
}
```

3.Video Preview

1) Turn on video preview

```
- (IBAction)play:(id)sender
{
    [[VSNet shareinstance] startLivestream:strDID withStream:10 withSubStream:2];
    [[VSNet shareinstance] setDataDelegate:strDID withDelegate:self];//设置代理接收图像数据
}
```

2) Turn on video preview

```
#pragma mark VSNetDataProtocol
- (void) VSNetYuvData: (NSString*) deviceIdentity data:(Byte *) buff withLen:(NSInteger)len
               height:(NSInteger)height width:(NSInteger)width time:(NSUInteger)timestame origenelLen:(NSInteger) oLen

{
    if ([deviceIdentity isEqualToString:strDID] == NO) {
        return;
    }

    if (myGLViewController) {
        SDL_VoutOverlay stOverlay;
        memset(&stOverlay, 0, sizeof(stOverlay));
        stOverlay.w = (int)width ;
        stOverlay.h = (int)height;
        stOverlay.pitches[0] = width;
        stOverlay.pitches[1] = stOverlay.pitches[2] = width /2;
        stOverlay.pixels[0] = buff;
        stOverlay.pixels[1] = buff + width*height;
        stOverlay.pixels[2] = buff + width*height*5/4;
        [myGLViewController display:&stOverlay];
    }
}
```

3) Turn off video preview

```
[[VSNet shareinstance] stopLivestream:strDID];
```

4.Sound control

1）Turn on listening

```
[[VSNet shareinstance] startAudio:strDID withEchoCancellationVer:NO];
```

2）Stop listening

```
[[VSNet shareinstance] stopAudio:strDID];
```

5.Intercom

1）Start conversation

```
[[VSNet shareinstance] startTalk:strDID withEchoCancellationVer:NO];
```

2）Stop conversation

```
[[VSNet shareinstance] stopTalk:strDID];
```

6.Search online devices in the LAN

1）Start searching

```
- (void) startSearch
{
    [[VSNet shareinstance] StartSearchDVS:self];
    //create the start timer
    searchTimer = [NSTimer scheduledTimerWithTimeInterval:2.0 target:self selector:@selector(handleTimer:) userInfo:nil repeats:NO];
}
```

2）Search for device callbacks

```
#pragma mark SearchCamereResultDelegate
- (void) VSNetSearchCameraResult:(NSString *)mac Name:(NSString *)name Addr:(NSString *)addr Port:(NSString *)port DID:(NSString*)did
{
    if ([did length] == 0) {
        return;
    }
    [searchListMgt AddCamera:mac Name:name Addr:addr Port:port DID:did];
}
```

3）Stop searching

```
[[VSNet shareinstance] StopSearchDVS];
```

7.Record video preview

1）Start recording a video

```
NSString* strBasePath = [self GetBasePath:strDID];
NSString* fileName = [strBasePath stringByAppendingPathComponent:@"test22.mp4"];
if (fileName != nil) {
    [[VSNet shareinstance] StartRecord:fileName cameraUid:strDID completion:^(BOOL success, int nError) {
        if (success) {
            NSLog(@"Record success");
            dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
```

2）Stop recording video

```
[[VSNet shareinstance] StopCameraUid:strDID];
```

8.SD Card recording

1）Get a list of SD card video files

```
    [[VSNet shareinstance] setControlDelegate:m_strDID withDelegate:self];
    [VSNetSendCommand VSNetCommandGetRecordFileWithDID:m_strDID user:@"admin" pwd:m_strPWD loginuse:@"admin" loginpas:m_strPWD pageSize:
        500 pageIndex:0];
```

2）Return a list of SD card video files

```
- (void)VSNetControl:(NSString *)deviceIdentity commandType:(NSInteger)comType buffer:(NSString *)retString length:(int)length charBuffer:
    (char *)buffer {
    NSLog(@"RemoteRecordFileListViewController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
    if (comType == CGI_IEGET_RECORD_FILE && [deviceIdentity isEqualToString:deviceIdentity]){
        [self performSelectorOnMainThread:@selector(StopTimer) withObject:nil waitUntilDone:YES];
        NSRange range = [retString rangeOfString:@"record_name0[0]="];
        if (range.location != NSNotFound)
        {
            NSInteger count = [[NSString subValueByKeyString:@"record_num0=" fromRetString:retString] integerValue];
            if (count > 0) {
               dispatch_async(dispatch_get_main_queue(), ^{
                   NSString *RecordCount = [NSString subValueByKeyString:@"RecordCount=" fromRetString:retString];
                   _recordCount = [RecordCount integerValue];
                   for (NSInteger i = 0; i < count; i ++) {
                       NSString* recordName = [NSString subValueByKeyString:[NSString stringWithFormat:@"record_name0[%ld]=",i]
                           fromRetString:retString];
                       NSString* recordSize = [NSString subValueByKeyString:[NSString stringWithFormat:@"record_size0[%ld]=",i]
                           fromRetString:retString];
```

Implicit conversion loses integer precision: 'NSInteger' (aka 'long') to 'int'

3）Play SD card video file

```
[[VSNet shareinstance] startPlayBack:strDID fileName:m_strFileName
withOffset:0 fileSize:_record_Size delegate:self SupportHD:1];
```

4）Stop playing SD card recording files

```
[[VSNet shareinstance] stopPlayBack:strDID];
```