

IOS VSNet 库使用说明

1.初始化库

1) XCODE: Enable Bitcode yes 改为 no

2) 依赖库



注意: XCode 10 以上时需要把依赖库 libstdc++.tbd 替换成 libc++.tbd.

3) 初始化库

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

2.设备管理

1) 连接设备

```
int nRet = [[VSNet sharedInstance] 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 sharedInstance] start:strDID withUser:@"admin" withPassWord:strPWD initializeStr:nil LanSearch:1];  
        [[VSNet sharedInstance] setStatusDelegate:strDID withDelegate:self]; //设置代理接收设备状态  
        [[VSNet sharedInstance] setControlDelegate:strDID withDelegate:self]; //设置代理接收所发指令设备回复  
    });  
}  
else{  
    [[VSNet sharedInstance] setStatusDelegate:strDID withDelegate:self];  
    [[VSNet sharedInstance] setControlDelegate:strDID withDelegate:self];  
}
```

2) 断开连接

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

3) 连接状态接收

```
#pragma mark VSNetStatueProtocol
- (void) VSNetStatus: (NSString*) deviceIdentity statusType:(NSInteger) statusType status:(NSInteger) status
{
    NSLog(@"PPPStatus ..... 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) 设备密码管理

(4.1) 重置设备密码

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

(4.2) 重置设备密码返回

```
- (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_ISET_USER && [deviceIdentity isEqualToString:m_strDID]) {
        NSInteger result = [[APICCommon stringAnalysisWithFormatStr:@"result=" AndRetString:retString] integerValue];
        if (result == 0){
            [[VSNet sharedInstance] 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) 设备 wifi 管理

(5.1) 获取当前设备 WIFI

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

(5.2) 获取当前设备 WIFI 返回

```
- (void) VSNetControl: (NSString*) deviceIdType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
charBuffer:(char *)buffer
{
    NSLog(@"WifiSettingViewController: VSNet返回数据 UID:%@,comType:%ld",deviceIdType,(long)comType); 2 Data argument not used by format stri...
    NSString *string = [[NSString alloc] initWithCString:buffer encoding:NSUTF8StringEncoding];
    if ([deviceIdType 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) 获取设备 WIFI 列表

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

(5.4) 获取设备 WIFI 列表返回

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

(5.5) 设置设备 WIFI

```
NSString *cmd = [NSString stringWithFormat:@"set_wifi.cgi?
enable=1&ssid=%@&encrypt=0&defkey=0&key1=%s&key2=%s&key3=%s&key4=%s&authType=%d&keyFormat=0&key1_bits=0&key2_bits=0&key3_bits=0&key4_bits=0&channel=%d&mode=0&wpa_psk=%s&",m_strSSID,pkey,m_security,m_channel,pwpa_psk];

NSString *sendSSID = [cmd stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];

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

6) 设备固件升级

```
NSLog(@"%@=++%@",self.firmware_server,self.firmware_file);
NSString *cmd = [NSString stringWithFormat:@"auto_download_file.cgi?
server=%@&file=%@&type=%d&reseed1=&reseed2=&reseed3=&reseed4=",self.firmware_server,self.firmware_file,0];
[[VSNet sharedInstance] sendCgiCommand:cmd withIdentity:self.str_uid];
```

7) 重启设备

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


8) 设备参数

(8.1) 获取设备参数

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

(8.2) 获取设备参数返回

```
4
5 #pragma mark - VSNetControlProtocol
6 - (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
  charBuffer:(char * )buffer
7 {
8     NSLog(@"DateTimeController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
9     if ( [deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM) {
10         m_timeZone = -[[APICommon stringAnalysisWithFormatStr:@"tz=" AndRetString:retString] integerValue];
11         m_dateTime = [[APICommon stringAnalysisWithFormatStr:@"now=" AndRetString:retString] integerValue];
12         m_timing = [[APICommon stringAnalysisWithFormatStr:@"ntp_enable=" AndRetString:retString] integerValue];
13         m_timingSever = [APICommon stringAnalysisWithFormatStr:@"ntp_svr=" AndRetString:retString];
14     }
```

9) 设备报警

(9.1) 获取报警参数

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

(9.2) 返回获取报警参数

```
9 #pragma mark - VSNetControlProtocol
0 - (void) VSNetControl: (NSString*) deviceIdentity commandType:(NSInteger) comType buffer:(NSString*)retString length:(int)length
  charBuffer:(char * )buffer
1 {
2     NSLog(@"AlarmController VSNet返回数据 UID:%@ comtype %ld",deviceIdentity,(long)comType);
3     if ( [deviceIdentity isEqualToString:m_strDID] && comType == CGI_IEGET_PARAM)
4     {
5         m_motion_armed = [[NSString subValueByKeyString:@"alarm_motion_armed=" fromRetString:retString] intValue];
6         m_motion_sensitivity = [[NSString subValueByKeyString:@"alarm_motion_sensitivity=" fromRetString:retString] intValue];
7         m_input_armed = [[NSString subValueByKeyString:@"alarm_input_armed=" fromRetString:retString] intValue];
8         m_ioin_level = [[NSString subValueByKeyString:@"alarm_ioin_level=" fromRetString:retString] intValue];
9         m_alarmpresetsit = [[NSString subValueByKeyString:@"alarm_presetsit=" fromRetString:retString] intValue];
0         m_iolinkage = [[NSString subValueByKeyString:@"alarm_iolinkage=" fromRetString:retString] intValue];
1         m_ioout_level = [[NSString subValueByKeyString:@"alarm_ioout_level=" fromRetString:retString] intValue];
2         m_mail = [[NSString subValueByKeyString:@"alarm_mail=" fromRetString:retString] intValue];
3         m_snapshot = [[NSString subValueByKeyString:@"alarm_snapshot=" fromRetString:retString] intValue];
4         m_upload_interval = [[NSString subValueByKeyString:@"alarm_upload_interval=" fromRetString:retString] intValue];
5         m_record = [[NSString subValueByKeyString:@"alarm_record=" fromRetString:retString] intValue];
6         m_enable_alarm_audio = [[NSString subValueByKeyString:@"enable_alarm_audio=" fromRetString:retString] intValue];
7         [self performSelectorOnMainThread:@selector(reloadTableView:) withObject:nil waitUntilDone:NO];
8     }
9 }
```

(9.3) 设置报警参数

```
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 sharedInstance] sendCgiCommand:cmd withIdentity:self.m_strDID];
```

10) 设备预置位

(10.1) 设置设备预置位 0

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

(10.2) 设置设备预置位 1

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

(10.3) 设置设备预置位 2

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

(10.4) 设置设备预置位 3

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

(10.5) 设置设备预置位 4

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

(10.6) 调用设备预置位

```
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 sharedInstance] sendCgiCommand:cgi withIdentity:_strDID];
}
```

11) 云台操作

(11.1) 上下巡航

```
- (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 sharedInstance] 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 sharedInstance] sendCgiCommand:cgi withIdentity:_strDID];
        btnUpDown.style = UIBarButtonItemStyleDone;
        [_upDownBtn setImage:_arrowUpDownImgOn forState:UIControlStateNormal];
    }
}
```


(11.2) 左右巡航

```
- (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 sharedInstance] 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 sharedInstance] sendCgiCommand:cgi withIdentity:strDID];

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

12) 图像传感器参数控制

(12.1) 翻转与镜像

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

(12.2) 亮度

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

(12.3) 对比度

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

13) 设备截图

(13.1) 获取设备截图

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

(13.2) 获取设备截图返回

```
# 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_ISET_SNAPSHOT:
        {
            NSData *image = [NSData alloc] initWithBytes:buffer length:length;
            [self SnapshotCallback:image UID:deviceIdentity];
            break;
        }
        default:
            break;
    }
}
```

3.预览视频

1)开启预览视频

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

2)开启预览视频

```
#pragma mark VSNetDataProtocol
- (void) VSNetYuvData: (NSString*) deviceIdty data:(Byte *) buff withLen:(NSInteger)len
    height:(NSInteger)height width:(NSInteger)width time:(NSUInteger)timestamp origenLen:(NSInteger) oLen
{
    if ([deviceIdty 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)关闭预览视频

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

4.监听声音

1) 开启监听

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

2) 停止监听

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

5.对讲

1) 开启对讲

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

2) 停止对讲

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

6. 局域网内搜索在线设备

1) 开始搜索

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

2) 搜索到设备回调

```
#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) 停止搜索

```
[[VSNet sharedInstance] StopSearchDVS];
```

7. 录制预览视频

1) 开始录制预览视频

```
NSString* strBasePath = [self GetBasePath:strDID];
NSString* fileName = [strBasePath stringByAppendingPathComponent:@"test22.mp4"];
if (fileName != nil) {
    [[VSNet sharedInstance] 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) 停止录制视频

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

8. SD 卡录像

1) 获取 SD 卡录像文件列表

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


2) 返回获取 SD 卡录像文件列表

```
- (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];
                    }
                });
            }
        }
    }
}
```

3) 播放 SD 卡录像文件

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

4) 停止播放 SD 卡录像文件

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