

RGB

snippet.cpp

```
//RGB YUV 가
char* pVideoBuffer = (char *)pData;

BITMAPINFOHEADER bmih;
bmih.biSize = sizeof(BITMAPINFOHEADER);
bmih.biWidth = nWidth;
bmih.biHeight = nHeight;
bmih.biPlanes = 1;
bmih.biBitCount = 24;
bmih.biCompression = BI_RGB;
bmih.biSizeImage = 0;
bmih.biXPelsPerMeter = 10;
bmih.biYPelsPerMeter = 10;
bmih.biClrUsed = 0;
bmih.biClrImportant = 0;

BITMAPINFO dbmi;
ZeroMemory(&dbmi, sizeof(dbmi));
dbmi.bmiHeader = bmih;
dbmi.bmiColors->rgbBlue = 0;
dbmi.bmiColors->rgbGreen = 0;
dbmi.bmiColors->rgbRed = 0;
dbmi.bmiColors->rgbReserved = 0;

HDC hdc = ::GetDC(NULL);

HBITMAP hbmp = CreateDIBitmap(hdc, &bmih, CBM_INIT, pVideoBuffer,
&dbmi, DIB_RGB_COLORS);

CWnd* pWnd = AfxGetApp()->GetMainWnd();
HWND hWnd = pWnd->GetSafeHwnd();
CStatic* m_staticPic2 = (CStatic*)pWnd->GetDlgItem(IDC_STATIC_PIC2);
m_staticPic2->SetBitmap(hbmp);
::ReleaseDC(NULL, hdc);
DeleteObject(hbmp);
```

:: AfxGetMainWnd()      AfxGetApp()->GetMainWnd()

- [\[MFC\] Raw](#)

## snippet.cpp

```
CImage img;
img.Create(nWidth, nHeight, 24);

int nPixel = 0;
for (int row = 0; row < nHeight; row++) {
    for (int col = 0; col < nWidth; col++) {
        BYTE r = pVideoBuffer[nPixel++];
        BYTE g = pVideoBuffer[nPixel++];
        BYTE b = pVideoBuffer[nPixel++];
        img.SetPixel(col, row, RGB(r, g, b));
    }
}

CBitmap* pBitmap = CBitmap::FromHandle(img);
CWnd* pWnd = AfxGetApp()->GetMainWnd();
HWND hWnd = pWnd->GetSafeHwnd();
CStatic* m_staticPic2 = (CStatic*)pWnd->GetDlgItem(IDC_STATIC_PIC2);
m_staticPic2->SetBitmap((HBITMAP)pBitmap->GetSafeHandle());
```

## snippet.cpp

```
FILE *fp;
static int nNum = 0;
char szFilename[1024];

sprintf(szFilename, "C:\\9.test\\img\\head%d.jpg", nNum);
fp = fopen(szFilename, "wb");
fwrite(pData, nDataSize, 1, fp);
fclose(fp);

sprintf(szFilename, "C:\\9.test\\img\\body%d.jpg", nNum);
fp = fopen(szFilename, "wb");
fwrite(pVideoBuffer, nDataSize - videoHeaderSize, 1, fp);
fclose(fp);
nNum++;
```

[snippet.cpp](#)

```
EXTERN_C INTEROPLIB_API bool W4NVMS_StartLiveVideo(int nDeviceID, int nStreamDataType)
```

—	—	—
Decompressed	- 1	Color Format(RGB24 or YUYV422)
RGB24	0	24 Bit RGB Format
YUV422	1	YUYV422 Format
JPEG	2	JPEG Format
H264	3	H264 Format
MPEG4	4	MPEG4 Format
ARGB	5	ARGB Format
Custom	32	
RAW	64	Encoding
MetaData	128	

int max : 2147483647  
1280 \* 1024 = 1310720

```
Sundance.Interop.AirLiveTestDlg.cpp(315) : atlTraceGeneral - ROW:308
:
'System.AccessViolationException' (Sundance.Interop.AirLiveTest.exe)
```

p4Ef7A

[snippet.cpp](#)

```
//videoStream가
char *pVideoBuffer = (char *) (pData + videoHeaderSize);
```

pVideoBuffer가 . nWidth \* nHeight.  
,

[snippet.cpp](#)

```
int size = nWidth * nHeight * nStreamDataType;
unsigned char* pImgData = new unsigned char[size];
memcpy(pImgData, pVideoBuffer, size);
```

, pdb

WR19nx

Connect

. DeviceId 44

StartVideo

snippet.cpp

```
void CALLBACK StreamDataCallback(int nDeviceID, int nWidth, int
nHeight, int nStreamDataType, LPBYTE pData, int nDataSize, wchar_t*
strTime, LPVOID pUserParam)
{
    int videoHeaderSize = sizeof(W4NVMS_StructRawVideoHeader);
    W4NVMS_StructRawVideoHeader* videoHeader =
(W4NVMS_StructRawVideoHeader *)pData;
    //videoStream 가
    char *pVideoBuffer = (char *) (pData + videoHeaderSize);
}
```

## Plugin Backlinks:

From:

<http://jace.link/> - Various Ways

Permanent link:

<http://jace.link/open/sundance-airlivetest-%EB%B6%84%EC%84%9D>

Last update: 2020/06/02 09:25

