

Samsung XNS ActiveX SDK

API Reference

XNS_ACTIVE_X_API_KR

Ed.1.44.2.0

2017-08-01

Copyright

©2011-2016 Hanwha Techwin Co., Ltd. All rights reserved.

Trademark

XNS ActiveX는 한화테크원(주)의 등록상표입니다. 이밖에 이 문서에 언급된 상표는 해당 회사 고유의 등록상표입니다.

Restriction

이 문서의 저작권은 한화테크원(주)에게 있습니다. 한화테크원(주)의 공식 승인없이 이 문서의 내용을 복제, 배포 및 재생산 할 수 없고, 임의로 변경할 수 없습니다.

Disclaimer

한화테크원(주)는 이 문서에 수록된 정보의 완결성과 정확성을 검증하기 위해 최대한 노력하였으나 이에 대해 보증하지는 않습니다. 문서의 사용 결과에 따른 책임은 전적으로 사용자에게 있습니다.

한화테크원(주)은 사전 예고 없이 이 문서의 내용을 변경 할 수 있습니다.

Contact Information

HANWHA TECHWIN Co., LTD.

Hanwha techwin R&D Center, 6, Pangyo-ro 319beon-gil, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea, 463-400
TEL: +82-70-7147-8740~60 FAX: +82-31-8018-3745

<https://step.hanwha-security.com>

SAMSUNG TECHWIN AMERICA Inc.

1480 Charles Willard St, Carson, CA 90746, UNITED STATES

Toll Free: +1-877-213-1222 FAX: +1-310-632-2195

SAMSUNG TECHWIN EUROPE LTD.

Samsung House, 1000 Hillswood Drive, Hillswood Business

Park Chertsey, Surrey, UNITED KINGDOM KT16 OPS

TEL: +44-1932-45-5300 FAX: +44-1932-45-5325

들어가며

목적

이 문서는 한화테크원(주)의 XNS ActiveX SDK가 제공하는 API 함수를 설명합니다. 사용자 애플리케이션 작성 시에 API 함수를 사용함으로써 당사 장비와 애플리케이션 간 인터페이스를 구현할 수 있습니다.

독자

이 문서는 XNS ActiveX SDK를 이용하여 애플리케이션 (CMS, 뷰어(viewer) 애플리케이션 등)을 구현하는 개발자를 독자로 합니다.

범위

이 문서는 XNS ActiveX SDK의 설치 방법, 애플리케이션 구현 방법 등에 대해 설명합니다.

문서 구성

이 문서는 다음과 같이 구성되어 있습니다.

CHAPTER 1. XNS ActiveX SDK: XNS ActiveX SDK를 소개합니다.

CHAPTER 2. XnsSdkDevice: XnsSdkDevice 컴포넌트에 속한 함수들에 대해 설명합니다.

CHAPTER 3. XnsSdkWindow: XnsSdkWindow 컴포넌트에 속한 함수들에 대해 설명합니다.

APPENDIX A. 에러 코드: XNS ActiveX 라이브러리에 정의된 에러 코드를 나열합니다.

APPENDIX C. UTC Time: UTC Time에 대해 설명합니다.

Convention

버튼 또는 메뉴 표기: 버튼명과 메뉴명은 대괄호([])를 사용해서 표기하였습니다.

메뉴 선택 경로 표기: 메뉴 선택 경로는 산괄호(>)를 사용해서 표기하였습니다.

문서 이력

이 문서의 출판 이력은 다음과 같습니다.

버전	개정일자	개정내역
0.9	2010. 08. 16	최초 작성
1.0	2010. 11. 25	API Reference 완료 (SDK v1.23)
1.1	2011. 05. 03	추가 설명 및 오류 수정. 디바이스 추가(SDK v1.30)
1.2	2011. 05. 25	Window 7에서 프로젝트 환경설정 GMT 시간 설정 추가
1.3	2011. 06. 16	Device Control 이용 관련 추가 설명 (SDK v1.30.06) API 추가 <ul style="list-style-type: none">• AreaZoom(), Zoom1X() 추가• OnIvEvent() 추가• GetPtzPos(), SetPtzPos() 추가• OnGetPtzPos() 추가
1.3.1	2011.6.24	SDK 아키텍처 변경 내용 반영 Stop() 예제 코드 수정 Start() 설명 수정
1.4.0	2011.08.05	API Reference에서 Programmer Guide 내용 분리 함수 추가 <ul style="list-style-type: none">• UTCToDeviceLocalTime() 추가 Event 추가 및 변경 <ul style="list-style-type: none">• OnIvEvent 명칭 변경 -> OnIvOccurEvent
1.4.1	2011.09.20	OpenStream() 리턴값 설명 수정
1.4.2	2011.09.29	AreaZoomMoving 이벤트 내용 추가
1.34.00	2011.11.02	문서 리비전 번호를 SDK 버전정보와 맞춤. 지원 제품 정보 업데이트
1.35.00	2012.01.06	XnsSdkWindow::Start() 함수에 주의사항 추가 에러코드 정보 업데이트
1.37.00	2012.03.09	XnsSdkDevice::Connect()에서 bForced 옵션에 대한 제약사항 추가 API 추가 <ul style="list-style-type: none">• XnsSdkWindow::SetVideoRawDataReceived()• XnsSdkWindow::UnsetVideoRawDataReceived()• XnsSdkWindow::OnVideoRawDataReceived()• XnsSdkWindow::SaveSnapshotWidthData()
1.38.00	2012.05.30	표 2.5 Capability ID 정보 업데이트

1.39.00	2012.08.31	<p>OnFanStatusChanged 이벤트 내용 추가 XnsSdkDevice::ExecuteHPtz() 내 SWING, TOUR 파라미터 설명 추가 OnNewMedia 이벤트에 주의사항 및 소스코드 내용 보강 지원장비 정보 변경 (모델명 통합)</p>
1.39.10	2012.12.03	<p>AVI Backup 관련 API 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::StartAviLocalRecording() • XnsSdkDevice::StopAviLocalRecording() • XnsSdkDevice::StartAviBackup() • XnsSdkDevice::StopAviBackup() <p>중복구간 검색 API 및 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::SearchOverLap() • XnsSdkDevice::OnSearchOverLap() <p>AutoScan, IPInstallDhcp 관련 API 및 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::IPInstallDhcp() • XnsSdkDevice::OnDeviceDetected2() • XnsSdkDevice::OnDeviceInstalled() <p>비디오 프레임 정보를 얻기 위한 SetVideoFrameInfoReceived API, OnVideoFrameInfoReceived 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::SetVideoFrameInfoReceived() • XnsSdkDevice::OnVideoFrameInfoReceived() <p>MotionGrid 관련 API 및 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetMotionGrid() • XnsSdkDevice::MotionSearch() • XnsSdkDevice::OnGetMotionGrid() • XnsSdkDevice::OnMotionSearch() <p>OnHDDNumChanged 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::OnHDDNumChanged() <p>녹화 시작/종료 시간 가져오는 API 및 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetStartRecordingTime() • XnsSdkDevice::GetEndRecordingTime() • XnsSdkDevice::OnGetStartRecordingTime() • XnsSdkDevice::OnGetEndRecordingTime() <p>OnRecordingErrorEvent 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::OnRecordingErrorEvent()
1.40.00	2013.03.29	<p>녹화상태 변경 이벤트 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::OnRecordingStatusChanged() <p>control module 의 Video 정보를 가져오는 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetVideoResolution() • XnsSdkDevice::GetVideoFrameRate() • XnsSdkDevice::GetVideoQuality() • XnsSdkDevice::GetVideoCodec()

1.40.10	2013.04.24	<p>Custom OSD API 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkWindow::SetCustomOSD() • XnsSdkWindow::SetCustomOSDOnOff() <p>Image Flip API 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkWindow::SetImageFlip() <p>Video pixel format API 내용 추가</p> <ul style="list-style-type: none"> • XnsSdkWindow::GetVideoPixelFormat()
1.40.30	2013.09.02	API 사용 제약사항에 대한 내용 추가
1.41.00	2014.03.28	<p>REC 파일을 AVI 파일로 변환하는 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::ConvertRec2Avi() • XnsSdkDevice::StopRec2Avi() <p>SendAudioData2 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::SendAudioData2() • XnsSdkDevice::StopAudioData2() <p>AVI 녹화 설정 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetAviOptionSize() • XnsSdkDevice::SetAviOptionSize() • XnsSdkDevice::GetAviOptionPeriod() • XnsSdkDevice::SetAviOptionPeriod()
1.41.10	2014.07.24	Supported device list 수정
1.41.20	2014.08.27	Contact Information 기술 지원 사이트 변경
1.41.70	2014.11.06	<p>이벤트 관련 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::OnTrackingEventChanged() • XnsSdkDevice::OnFaceDetectionEventChanged() • XnsSdkDevice::OnAudioDetectionEventChanged() • XnsSdkDevice::OnAMDEventChanged() • XnsSdkDevice::OnSystemEventChanged() <p>XNS_AUDIO_HEADER2 설명 추가</p> <ul style="list-style-type: none"> • OnNewMedia() <p>GetDeviceID(), GetDeviceStatus() 리턴값 설명 수정</p> <p>접속 정보 설정 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::SetConnectionInfo2() <p>OnDeviceDetected3 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::OnDeviceDetected3()
1.41.90	2014.12.15	<p>SaveSnapshotWithString API 추가</p> <ul style="list-style-type: none"> • XnsSdkWindow::SaveSnapshotWithString()
1.42.30	2015.03.06	<p>DVR/NVR 시간 정보 얻기 위한 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetDeviceUTC() • XnsSdkDevice::GetDeviceLocalTime() <p>카메라 비밀번호 변경 관련 API 추가</p>

		<ul style="list-style-type: none"> • XnsSdkDevice::ChangeDeviceAdminPassword() • XnsSdkDevice::OnChangeDeviceAdminPassword() <p>OnNewMedia()에 Metadata 추가</p>
1.42.40	2015.03.31	<p>SEC 암호화 백업 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::StartSecBackup()
1.42.50	2015.08.24	<p>PTZ 좌표 정보 관련 API 및 이벤트 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetPtzPosNormalize() • XnsSdkDevice::SetPtzPosNormalize() • XnsSdkDevice::OnGetPtzPosNormalize()
1.42.90	2015.11.12	<p>종횡비 유지 관련 API 추가</p> <ul style="list-style-type: none"> • XnsSdkWindow::SetAspectRatio() <p>표 1.1 Supported Device List 수정 표 2.1 Device Model Name 추가 DEP 문제 해결 방법 추가</p>
1.43.20	2016.03.25	<p>ControlID 얻는 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetVideoControlID() <p>그림 2.2 멀티프로파일 지원하는 DVR/NVR 의 Control Module 구조 추가</p>
1.44.00	2016.07.29	<p>NVR POS 연동 API 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::SearchTextCalendar() • XnsSdkDevice::SearchText () • XnsSdkDevice::CancelSearchText() • XnsSdkDevice::OpenText() • XnsSdkDevice::CloseText() • XnsSdkDevice::GetControlLinkStatus() • XnsSdkDevice::OnTextOccur() • XnsSdkDevice::OnSearchTextCalendar() • XnsSdkDevice::OnSearchTextList() • XnsSdkDevice::OnEndSearchText ()
1.44.20	2017.08.01	<p>Video End Event 추가</p> <ul style="list-style-type: none"> • XnsSdkDevice::GetVideoControlID()

목차

들어가며.....	2
문서 이력.....	3
목차.....	7
그림 목차.....	18
표 목차.....	19
CHAPTER 1 XNS ActiveX SDK.....	20
개요.....	21
SDK Architecture	21
SDK 구성.....	23
지원 장비.....	24
CHAPTER 2 XnsSdkDevice.....	24
Initialize.....	26
Initialize	26
GetConfigPath.....	27
Find.....	27
FindVendor.....	28
FindModel.....	29
FindDevice	30
GetFindSize	30
GetFindString	31
GetFindLong	31
GetFindDouble	33
CloseFind.....	34
Create/Release Device	34
CreateDevice.....	35

ReleaseDevice	35
Get/Set Device Information	36
SetConnectionInfo	36
SetConnectionInfo2	39
GetDeviceID	40
GetDeviceStatus	41
GetVendorName	42
GetModelName	42
GetVersion	43
GetAddressType	44
GetIP	44
GetMac	45
GetURL	46
GetPort	46
GetHttpPort	47
GetUserID	48
GetPassword	48
GetLatestUpdateTime	49
GetDeviceHandle	49
GetDeviceUTC	50
GetDeviceLocalTime	51
ChangeDeviceAdminPassword	51
Connect/Disconnect	52
Connect	52
ConnectNonBlock	54
Disconnect	55
Control Module Information	56
GetControlCount	59
GetControlType	59
GetControlNumber	60
GetControlDepth	61
GetControlLocalName	62

GetControlRemoteName	63
GetControlStatus	64
GetControlCapability	66
ChangeControlLocalName	71
GetControlLinkStatus	71
GetVideoControlID	73
GetParentControlID	73
GetVideoResolution	74
GetVideoFrameRate	75
GetVideoQuality	76
GetVideoCodec	77
Start/End to Play Media	78
AcquireMediaAuthority	78
ReleaseMediaAuthority	79
OpenMedia	79
OpenMediaEx	81
CloseMedia	82
OpenStream	83
CloseStream	84
SuspendStream	85
SuspendAllStream	86
ResumeStream	87
ResumeAllStream	88
IsPlaying	89
HasAuthority	90
SetVideoFrameInfoReceived	91
OpenText	92
CloseText	93
PlayControl	94
Play	95
Pause	96
Seek	96
FrameAdvance	97

SearchCalendar.....	98
SearchOverLap.....	99
GetStartRecordingTime	101
GetEndRecordingTime	101
SearchDay.....	102
GetMotionGrid	104
MotionSearch.....	105
SearchTextCalendar	107
SearchText.....	108
CancelSearchText.....	110
Control Audio and Alarm-out	111
SetTalk.....	111
SetListen	112
SendAudioData	113
SendAudioData2.....	114
StopAudioData2.....	115
SetAlarm	115
Control PTZ.....	117
ControlPtz.....	117
ControlPtzFreeMove	118
ControlPtzSpeedZoomIn	120
ControlPtzSpeedZoomOut	121
ControlPPtz	122
ControlMenu	123
ExecuteHPtz	124
GetPresetList.....	126
SetPreset.....	127
DeletePreset.....	128
DeletePresetAll	129
GetPresetCount.....	130
GetPreset.....	131
GetPtzPos.....	132
SetPtzPos.....	133

GetPtzPosNormalize	135
SetPtzPosNormalize	136
AreaZoom	137
Zoom1X	139
Timeline	140
GetCameraCount	141
GetTimelineCount	141
GetTimeline	143
Recording	146
StartLocalRecording	146
StopLocalRecording	147
StartAviLocalRecording	148
StopAviLocalRecording	149
ToggleDvrRecording	150
Backup	150
StartBackup	150
StopBackup	152
StartAviBackup	153
StopAviBackup	155
GetAviOptionSize	155
SetAviOptionSize	156
GetAviOptionPeriod	156
SetAviOptionPeriod	157
StartSecBackup	157
CreateFileReader	159
ReleaseFileReader	160
GetFileName	161
GetMediaSource	161
GetTimelineHandle	162
PlayReader	163
PauseReader	163
ConvertRec2Avi	164

StopRec2Avi	165
Update Firmware and Import/Export Configuration.....	165
UploadSoftware.....	165
ImportConfig.....	166
ExportConfig.....	167
Device Profile.....	168
LoadDeviceProfile.....	169
SaveDeviceProfile	170
DeleteDeviceProfile	170
XML File.....	171
ReadConfigValue	171
WriteConfigValue	172
DeleteConfigValue	173
CopyXml	173
FindConfigChildeKeyList	174
Miscellaneous	175
DateToTimet.....	175
TimetToDate.....	176
LocalTimeToDeviceUTC.....	178
UTCToDeviceLocalTime.....	179
AutoScan	179
IPInstall.....	180
IPInstallDhcp.....	182
SetEventDiscardTime	183
GetErrorString	184
PowerRestart.....	184
Events	185
OnControlLocalNameChanged.....	185
OnDeviceDetected.....	186
OnDeviceDetected2	188
OnDeviceDetected3	190
OnDeviceInstalled	192

OnConnectFailed	194
OnSearchCalendar	195
OnSearchTextCalendar	197
OnSearchOverLap	198
OnGetStartRecordingTime	199
OnGetEndRecordingTime	200
OnSearchDay	201
OnGetMotionGrid	203
OnMotionSearch	205
OnSearchTextList	206
OnEndSearchText	208
OnGetPresetList	209
OnUploadSoftwareProgress	210
OnImportConfigProgress	211
OnExportConfigProgress	212
OnDeviceStatusChanged	213
OnControlStatusChanged	214
OnControlCapabilityChanged	215
OnMotionEvent	217
OnSensorEvent	218
OnIvOccurEvent	219
OnVideoLossEvent	220
OnVideoLossEnd	221
OnTextOccur	223
OnHDDFailed	224
OnHDDFulled	224
OnHDDNumChanged	225
OnPasswordChanged	226
OnBackupProgress	227
OnNewMedia	228
OnFanBroken	234
OnFanStatusChanged	234
OnPtzControlFailed	235

OnAlarmOutControlFailed.....	236
OnControlRemoteNameChanged	237
OnGetPtzPos.....	238
OnGetPtzPosNormalize	239
OnAreaZoomMoving.....	240
OnVideoFrameInfoReceived.....	241
OnRecordingErrorEvent.....	243
OnPowerRestart	244
OnRecordingStatusChanged.....	245
OnTrackingEventChanged.....	246
OnFaceDetectionEventChanged.....	248
OnAudioDetectionEventChanged	249
OnAMDEventChanged.....	251
OnSystemEventChanged	253
OnChangeDeviceAdminPassword	256

CHAPTER 3 XnsSdkWindow 258

Initialize the Control.....	260
Initialize	260
Start and Stop	261
Start	261
Stop.....	262
Get Device Information	262
GetDeviceID	262
GetControlID.....	263
GetImageWidth.....	263
GetImageHeight	264
IsMedia.....	264
GetMediaType.....	265
Control Zoom	266
ZoomLevel	266
ZoomRate	267
GetZoomInfo	268

ClearZoom.....	269
DrawRect.....	270
ClearDraw.....	271
DrawShape	271
OSD Menu.....	273
SetOSDOnOff	273
SetDateTimeFormat	274
SetCustomOSD.....	275
SetCustomOSDOnOff.....	276
Contorl Volume.....	276
SetSound.....	276
SetVolume.....	277
GetVolume.....	278
Video Raw Data.....	278
SetVideoRawDataReceived	278
UnSetVideoRawDataReceived	279
Snapshot	280
SaveSnapshot.....	280
SaveSnapshotWithString.....	281
SaveSnapshotWithData.....	282
PrintSnapshot.....	284
Control Playback	284
PushMedia.....	284
SetPlaySpeed.....	286
GetPlaySpeed.....	286
MoveFrameAdvance	287
Control Buffer.....	287
ClearMediaBuffer.....	288
IsBufferFull.....	288
IsBufferEmpty	289
SetBufferSize.....	289
GetBufferSize.....	290

GetRemainingBufferSize.....	291
Get Display Information	291
GetCurrentFrameTime.....	291
GetFirstFrameTime.....	292
GetInputType – 사용하지 않음.....	293
GetFrameType – 사용하지 않음	293
GetFrameTime – 사용하지 않음.....	294
GetFrameSize – 사용하지 않음	295
GetTimeStamp - 사용하지 않음.....	295
SetBrightness.....	296
GetBrightness.....	297
SetContrast.....	297
GetContrast.....	298
SetDeinterlaceMode	298
SetDefogMode	299
SetImageFlip.....	299
GetVideoPixelFormat.....	300
SetAspectRatio	301
Miscellaneous	301
DisplayNoData.....	302
SetCursor.....	302
GetLastError	303
DateToTimet.....	303
TimetToDate.....	304
Events	306
OnFrameTimeChanged.....	306
OnImageResolutionChanged.....	307
OnMediaOn	307
OnMediaOff	308
OnVideoRawDataReceived	308
OnLButtonDown	311
OnLButtonUp	312

OnRButtonDown.....	313
OnRButtonUp.....	314
OnLButtonDblClk.....	315
OnRButtonDblClk.....	316
OnMouseMove.....	317
OnMouseHover.....	318
OnMouseLeave.....	318
OnBufferFull.....	319
OnBufferEmpty.....	320
OnPreBufferFull	320
OnPreBufferEmpty	321
OnEventStatusChanged.....	321
OnKeyDown.....	322
OnKeyUp	323
APPENDIX A DEP 문제 해결 방법.....	326
APPENDIX B 예러코드.....	329
APPENDIX C UTC Time.....	332
용어.....	335
약어.....	336
색인.....	337
OPEN SOURCE LICENSE REPORT ON THE PRODUCT	341

그림 목차

그림 1.1 XNS ActiveX Architecture	22
그림 2.1 DVR(SRD-1670)의 Control module 구조.....	56
그림 2.2 멀티프로파일 지원하는 DVR/NVR Control module 구조.....	57
그림 2.3 네트워크 카메라(SNB-5000)의 Control module 구조.....	57
그림 3.1 Zoom Rectangle의 예	267

표 목차

표 1.1 Supported Device List.....	24
표 2.1 Device Model Name	오류! 책갈피가 정의되어 있지 않습니다.
표 2.2 주소 타입에 따른 실제 주소값.....	38
표 2.3 Control module type에 따른 capability.....	58
표 2.4 Status ID.....	65
표 2.5 GetControlStatus()의 Return Value.....	65
표 2.6 Capability ID	67
표 2.7 PTZ Command.....	117
표 2.8 Menu Control Command	123
표 2.9 Recording Type	144
표 2.10 Device Information in device.xml	169
표 2.11 Control Module Status.....	215
표 2.12 Control Module Capability	216
표 2.13 AMD SubEvent List	252
표 2.14 System SubEvent List.....	254
표 3.1 OSD Type	273
표 3.2 Event Status	322
표 3.3 OnKeyDown 이벤트의 nFlags가 나타내는 항목	323
표 3.4 OnKeyUp 이벤트의 nFlags가 나타내는 항목	324

CHAPTER 1

XNS ActiveX SDK

이 장은 XNS ActiveX SDK의 개요, 구성 및 설치/삭제 방법 등에 대해 설명합니다.

Contents

- 개요
- SDK Architecture
- SDK 구성
- 시스템 요구 사양
- 지원 장비
- SDK 설치
- SDK 삭제

개요

XNS ActiveX SDK 는 XNS 라이브러리를 기반으로 Microsoft의 ActiveX 기술을 채택하여 만든 라이브러리입니다. ActiveX 기술을 채택함으로써 XNS 라이브러리를 C#, C++, Visual Basic, Internet Explorer용 웹 페이지(Java script) 등 다양한 플랫폼과 언어에서 개발하는 것이 가능해졌습니다.

XNS 라이브러리는 저수준(low-level)의 스트리밍과 이벤트 수신을 지원하는 반면, XNS ActiveX 라이브러리는 영상을 표현하고 음성 데이터를 재생할 수 있습니다. XNS ActiveX 라이브러리의 개발 목적은 개발자가 삼성의 네트워크 장비(예, 카메라, DVR, NVR)를 쉽게 통합할 수 있게 하기 위함입니다.

XNS ActiveX 라이브러리를 이용하여 애플리케이션을 구현하기 위해서는 XNS ActiveX SDK에서 제공하는 DLL 파일과 ActiveX 파일이 필요하며, 특히 ActiveX 파일 (즉, ocx 파일)은 레지스트리에 등록해야 합니다. XNS ActiveX SDK의 설치 파일을 이용하면 DLL파일과 ActiveX 파일이 지정된 위치에 저장됩니다. 설치 방법은 'XNS ActiveX Programmer's Guide' 문서를 참조하기 바랍니다.

XNS ActiveX 라이브러리에서 제공하는 기능은 다음과 같습니다.

- 실시간 영상 감시
- 네트워크 장비에 저장된 영상 재생 및 정보 획득
- 로컬 PC에 영상 저장 및 로컬 PC에 저장된 영상 재생
- 네트워크 장비에 저장된 영상을 로컬 PC로 백업(back-up)
- 네트워크 장비의 펌웨어(firmware) 업그레이드
- 네트워크 장비(카메라)의 PTZ 기능 제어
- 네트워크 장비에서 발생한 이벤트 정보 획득

SDK Architecture

XNS ActiveX SDK의 구조는 다음과 같습니다.

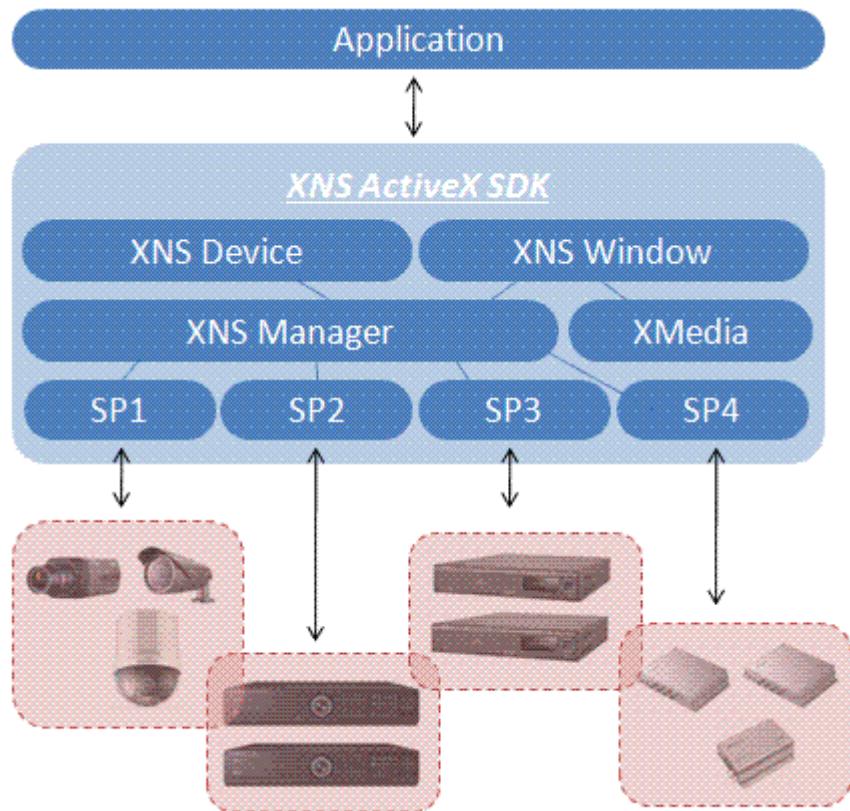


그림 1.1 XNS ActiveX Architecture

- Application
 - DVR이나 네트워크 카메라 같은 네트워크 장비에 있는 영상 데이터를 감시, 재생, 검색하는 사용자 프로그램.
- XNS ActiveX SDK
 - XNS Device: 네트워크 장비와 XNS 라이브러리를 관리하는 ActiveX control.
 - XNS Window: 비디오/오디오 데이터를 재생하는 ActiveX control.
 - XNS Manager: XNS 라이브러리와 이벤트를 관리하는 라이브러리(DLL).
 - XMedia: 수신된 영상/오디오 데이터를 디코딩&렌더링함
 - SP 라이브러리: 네트워크 장비로부터 받은 영상 데이터를 XNS Manager로 전달하고 사용자 명령에 따라 장비를 제어함.
- 네트워크 장비(DVR, NVR, Network camera)
 - 특정 영역에서 수집한 영상 데이터를 다른 장비로 전송함.

SDK 구성

XNS ActiveX SDK는 다음과 같이 구성되어 있습니다.

- 라이브러리 및 ActiveX control
 - 애플리케이션 개발 시 사용되는 DLL 파일과 OCX 파일.
 - Bin 폴더에 설치됨.
- 설정 파일 (Configuration file)
 - 시스템 설정을 위한 XML 파일은 {\$SDK path}\Bin\Config 폴더에 설치됨.
 - 특히, Xns.xml 파일은 각 프로토콜에 대한 DLL 파일의 정보를 저장하고 있음.
- 프로토콜 구현 파일 (DLL)
 - {\$SDK path}\Bin\SP 폴더에 설치됨.
 - 이 파일의 위치가 변경되는 경우, Xns.xml 문서의 내용도 수정되어야 함.
- Simple backup file player
 - {\$SDK path}\Bin\Viewer 폴더에 설치됨.
 - 애플리케이션이 미디어 스트림을 'SEC' 포맷으로 저장/백업한 경우, 저장된 파일은 이 플레이어를 이용하여 재생할 수 있음. 이때, 뷰어의 파일명은 라이브러리에서 백업 파일과 동일한 이름을 부여함. 뷰어를 실행하면, 같은 이름을 가진 SEC 파일이 자동으로 로딩/loading)됨.
- 헤더(header) 파일
 - {\$SDK path}\sample_code\include 폴더에 설치됨.
 - 애플리케이션이 XNS를 사용하기 위해서는 이 헤더 파일들을 인클루드(include)해야 함.
- 문서
 - {\$SDK path}\Doc 폴더에 저장됨.
- 샘플 코드
 - {\$SDK path}\sample_code 폴더에 저장됨.
 - 영상 데이터를 감시/재생하는 샘플과 PTZ를 제어하는 샘플을 제공함.
 - 샘플 코드의 실행 파일은 {\$SDK path}\Bin 폴더에 설치됨.

지원 장비

XNS ActiveX SDK를 지원하는 장비 리스트는 다음과 같습니다.

참고

모든 장비가 SDK의 모든 함수를 사용할 수 있는 것은 아닙니다. 사용하시는 장비가 해당 API를 지원하는지 먼저 확인하시기 바랍니다. 사용자는 장비에 연결한 후 GetControlCapability() 함수를 사용하여 해당 장비의 지원 기능을 확인할 수 있습니다.

표 1.1 Supported Device List

Device Type	Supported Device List
N/W Camera Encoder	현재 시판중인 네트워크 카메라, 엔코더 전모델 *홈 네트워크 카메라 제외
DVR	현재 시판중인 DVR 전모델 *PC DVR(SPR-XXXX) 제외
NVR	현재 시판중인 NVR 전모델 *SRN-872 제외

CHAPTER 2

XnsSdkDevice

XnsSdkDevice는 네트워크 장비와 XNS 라이브러리 간의 연결을 관리하고 네트워크 장비를 제어하는 컴포넌트입니다.

Contents

Initialize

Find

Create/Release Device

Get/Set Device Information

- Connect/Disconnect
- Control Module Information
- Start/End to Play Media
- Play Control
- Control Audio and Alarm-out
- Control PTZ
- Timeline
- Recording
- Backup
- Update Firmware and Import/Export Configuration
- Device Profile
- XML File
- Miscellaneous
- Events

Initialize

Initialize

Description

DLL 파일을 초기화합니다. 이를 위해서, XnsActiveX 라이브러리는 config.xml, device.xml, xns.xml 파일이 필요하며 DLL 파일의 리스트는 Xns.xml 파일에 기술되어 있어야 합니다. DLL 파일의 경로의 길이는 512바이트를 넘을 수 없습니다. XnsActiveX 라이브러리는 {\$SDK path}₩Config 폴더에 설치된 XnsSDKDevice.ocx 파일을 통해서 xns.xml 파일을 검색합니다.

Syntax

```
long Initialize(void);
```

Return Value

이 함수 수행이 성공하면 ERR_SUCCESS를, 실패하면 해당 에러 코드를 리턴합니다.

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_CFG_FILE: 잘못된 Xns.xml 파일.
- ERR_ALREADY_INITIALIZE: 라이브러리가 이미 초기화 됨.

Remarks

- Xns.xml에 기술된 DLL파일을 로딩/loading)하는 것이 실패하면, 에러 메시지 팝업이 화면에 표시됩니다. 에러 메시지에는 device ID가 포함되어 있습니다.
- XNS ActiveX Device Control은 한 프로세스 내 하나만 존재하여야 하며, 따라서 Initialize 작업 또한 한 번만 수행해야 합니다.
- Initialize 실패 시, '**APPENDIX A**' 참조

Example

[C++]

```
long ErrCode = m_ctlXnsDevice.Initialize();
{
    char str[100];
    sprintf(str, "ErrorCode[0x%x]\n", ErrCode);
    OutputDebugString(str);
}
```

See Also

[XnsSdkWindow::Initialize](#)

GetConfigPath

Description

Configuration 폴더 (즉, {\$SDK path}\Bin\Config)의 전체 경로를 반환합니다. Xns.xml 파일은 이 함수가 반환한 경로에 반드시 있어야 합니다.

Syntax

```
BSTR GetConfigPath(void);
```

Return Value

Configuration 폴더의 전체 경로를 반환합니다.

Example

[C++]

```
CString str = m_ctlXnsDevice.GetConfigPath();
```

Find

설정 파일에 기술된 정보를 검색하여 반환합니다. 이 기능을 사용하기 위해서는 다음의 순서를 따르도록 합니다.

절차

- Step 1. FindVendor(), FindModel(), FindDevice() 등을 이용하여 검색을 시작함.
- Step 2. GetFindSize()를 이용하여 배열의 크기를 구함.
- Step 3. GetFindString()를 이용하여 벤더(vendor)이름과 모델명을 얻음.
- Step 4. GetFindLong()과 GetFindDouble() 등을 이용하여 device ID를 얻음. (이 과정은 필요에 따라 반복적으로 수행)

Step 5. CloseFind()를 호출하여 검색을 종료함.

Example

[C++]

```
long hFind = m_ctlXnsDevice.FindModel("Samsung");
long nFindSize = m_ctlXnsDevice.GetFindSize(hFind);

CString str;
for (int i=0; i<nFindSize; i++)
{
    str = m_ctlXnsDevice.GetFindString(hFind, i);
    OutputDebugString(str);
}
m_ctlXnsDevice.CloseFind(hFind);
```

[C#]

```
int hFind = this._axXnsDevice.FindVendor();
int nFindSize = this._axXnsDevice.GetFindSize(hFind);

string[] VendorList = new string[nFindSize];
for (int i=0; i<nFindSize; i++)
{
    VendorList[i] = this._axXnsDevice.GetFindString(hFind, i);
}
this._axXnsDevice.CloseFind(hFind);
```

FindVendor

Description

Xns.xml파일에 기술된 벤더 리스트(vendor list)를 반환합니다.

Syntax

long FindVendor(void);

Return Value

Find 배열의 핸들을 반환합니다. 벤더 리스트가 없는 경우, NULL(0)이 리턴됩니다.

Example

[C++]

```
long hFind = m_ctlXnsDevice.FindVendor();
long nFindSize = m_ctlXnsDevice.GetFindSize(hFind);

CString str;
for (int i=0; i<nFindSize; i++)
{
    str = m_ctlXnsDevice.GetFindString(hFind, i);
    OutputDebugString(str);
}
m_ctlXnsDevice.CloseFind(hFind);
```

See Also

[FindModel](#), [FindDevice](#)

FindModel

Description

Xns.xml파일에 기술된 모델 리스트(model list)를 반환합니다.

Syntax

```
long FindModel(BSTR szVendorName);
```

Parameters

szVendorName

[in] 벤더명(vendor name). 이 값은 "Samsung"이 되어야 함. (대소문자 관계 없음.)

Return Value

Find 배열의 핸들을 반환합니다. 모델 리스트가 없는 경우, NULL(0)이 리턴됩니다.

Example

'Find'절의 예제를 참고하시기 바랍니다.

See Also

[FindVendor](#), [FindDevice](#)

FindDevice

Description

장비 리스트 파일(device list file)에 저장된 모든 장비의 ID를 반환합니다. 이 파일은 애플리케이션이 SaveDeviceProfile()을 이용하여 장비의 정보를 저장할 때, 생성됩니다.

Syntax

```
long FindDevice(void);
```

Return Value

Find 배열의 핸들을 반환합니다.

Example

'Find'절의 예제를 참고하시기 바랍니다.

See Also

[FindVendor](#), [FindModel](#), [SaveDeviceProfile](#)

GetFindSize

Description

FindVendor(), FindModel(), FindDevice() 등에 의해 생성된 배열의 길이를 반환합니다.

Syntax

```
long GetFindSize(long hFind);
```

Parameters

hFind

[in] FindXXX()에서 반환된 핸들.

Return Value

FindVendor(), FindModel(), FindDevice() 등에 의해 생성된 배열의 길이를 반환합니다.

Example

'Find'절의 예제를 참고하시기 바랍니다.

See Also

[FindVendor](#), [FindModel](#), [FindDevice](#)

GetFindString

Description

FindXXX() 수행 시 생성되는 문자열 데이터를 읽어오는 함수입니다. 즉, FindVendor() 또는 FindModel()을 호출하면 핸들(*hFind*)이 리턴되고 내부적으로 문자열 데이터를 갖는 배열이 생성되는데, 이 배열에 저장된 문자열 데이터는 이 함수를 이용하여 읽어올 수 있습니다.

Syntax

```
BSTR GetFindString(  
    long hFind,  
    long nIndex  
)
```

Parameters

hFind

[in] FindXXX()에서 반환된 핸들.

nIndex

[in] 데이터를 가져올 배열의 인덱스 번호.

Return Value

FindVendor(), FindModel() 등에 의해 생성된 배열에서 *nIndex*의 데이터를 반환합니다. *nIndex*가 실제 배열의 크기보다 큰 경우, ""이 리턴됩니다.

Example

'Find'절의 예제를 참고하시기 바랍니다.

See Also

[FindVendor](#), [FindModel](#), [FindDevice](#)

GetFindLong

Description

FindDevice() 수행 시 생성된 배열에서 장비 ID를 읽어오는 함수입니다. 즉, FindDevice()를 호출하면 핸들(*hFind*)이 리턴되고 내부적으로 장비 ID를 갖는 배열이 생성되는데, 이 배열에 저장된 정수형 데이터는 이 함수를 이용하여 읽어올 수 있습니다.

Syntax

```
long GetFindLong(
    long hFind,
    long nIndex
);
```

Parameters

hFind

[in] FindXXX()에서 반환된 핸들.

nIndex

[in] 데이터를 가져올 배열의 인덱스 번호.

Return Value

FindDevice()에 의해 생성된 배열에서 *nIndex*의 데이터를 반환합니다. *nIndex*가 실제 배열의 크기보다 큰 경우, -1이 리턴됩니다.

Example

[C++]

```
long hFind = m_ctlXnsDevice.FindDevice();
long nFindSize = m_ctlXnsDevice.GetFindSize(hFind);
long data;

for (int i=0; i<nFindSize; i++)
{
    data = m_ctlXnsDevice.GetFindLong(hFind, i);

    char str[100];
    sprintf(str, "data[%d] = [%d]\n", i, data);
    OutputDebugString(str);
}

m_ctlXnsDevice.CloseFind(hFind);
```

See Also

[FindDevice](#)

GetFindDouble

Description

FindXXX() 수행 시 생성되는 배열에서 Double형 데이터를 읽어오는 함수입니다.

Syntax

```
double GetFindDouble(
    long hFind,
    long nIndex
);
```

Parameters

hFind

[in] FindXXX()에서 반환된 핸들.

nIndex

[in] 데이터를 가져올 배열의 인덱스 번호.

Return Value

FindXXX()에 의해 생성된 배열에서 *nIndex*의 데이터를 반환합니다. *nIndex*가 실제 배열의 크기 보다 큰 경우, -1이 리턴됩니다.

Remarks

현재 FindXXX()에 의해서 생성된 배열 중 double형 데이터를 갖는 배열은 없습니다. 따라서, 이 함수는 향후에 사용될 예정입니다.

Example

[C++]

```
long hFind = m_ctlXnsDevice.FindDevice();
long nFindSize = m_ctlXnsDevice.GetFindSize(hFind);
double data;

for (int i=0; i<nFindSize; i++)
{
    data = m_ctlXnsDevice.GetFindDouble(hFind, i);
```

```

{
    char str[100];
    sprintf(str, "data[%d] = [%f]\n", i, data);
    OutputDebugString(str);
}
}

m_ctlXnsDevice.CloseFind(hFind);

```

See Also

[FindDevice](#)

CloseFind

Description

Find 명령을 종료합니다. FindXXX()에 의해서 할당된 메모리를 해제합니다.

Syntax

```
void CloseFind(long hFind);
```

Parameters

hFind

[in] FindXXX()에서 반환된 핸들.

Example

[C++]

```
m_ctlXnsDevice.CloseFind(hFind);
```

See Also

[FindVendor](#), [FindModel](#), [FindDevice](#)

Create/Release Device

CreateDevice

Description

장비의 정보를 저장하기 위한 메모리 공간을 할당하는 함수입니다. 이 함수가 수행되면 장비에 대한 핸들(device handle)이 반환되며, 애플리케이션은 이를 이용하여 장비를 제어할 수 있습니다.

Syntax

```
long CreateDevice(long nDeviceID);
```

Parameters

nDeviceID

[in] Device ID. 이 값은 0보다 큰 정수이어야 함.

최소값: 1, 최대값: 3000

Return Value

메모리가 정상적으로 할당되면, 장비에 대한 핸들을 리턴합니다. 그렇지 않을 경우, NULL(0)이 리턴됩니다.

Example

[C++]

```
long m_nDevice;  
m_hDevice = m_ctlXnsDevice.CreateDevice(nDeviceID);
```

See Also

[GetDeviceID](#)

ReleaseDevice

Description

CreateDevice()에서 장비 정보를 저장하기 위해 할당한 메모리를 해제합니다. 해당 장비는 이 함수가 호출되기 전에 애플리케이션과 연결이 해제된 상태여야 합니다.

Syntax

```
void ReleaseDevice(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Example

[C++]

```
long m_hDevice;

m_hDevice = m_ctlXnsDevice.CreateDevice(nDeviceID);

// Place your code here.

m_ctlXnsDevice.ReleaseDevice(m_hDevice);
```

See Also

[CreateDevice](#)

Get/Set Device Information

SetConnectionInfo

Description

애플리케이션이 장비에 접속하기 위해 필요한 장비 정보를 설정하는 함수입니다.

Syntax

```
void SetConnectionInfo(
    long hDevice,
    BSTR szVendorName,
    BSTR szmodelName,
    long nAddressType,
    BSTR szAddress,
    long nPort,
```

```

        long nHttpPort,
        BSTR szUserID,
        BSTR szPassword
);

```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

szVendorName

[in] 'Samsung'으로 고정임. (최대 길이는 126 바이트)

szModelName

[in] 접속할 장비의 통합 모델명. (최대 길이는 126바이트)

표 2.1 Device Model Name

Device Type	Model Name
Network Camera/Encoder	Samsung Network Camera/Encoder
DVR	Samsung DVR
NVR	Samsung NVR
Exceptions	*아래 기재된 제품은 모델명 입력 - Camera/Encoder : SNB-1000, SNB-2000, SNB-3000, SND-3080, SND-3080C, SND-3080F, SND-3080CF, SNV-3080, SNC-B2315, SNC-B2331, SNC-B2335, SNC-B5368, SNC-B5395, SNC-C6225, SNC-C7225, , SNC-C7478, SNC-M300, SNV-3370, SNP-3370TH, SNP-3301, SNP-3301H, SNC-550, SNC-570, SNC-1300, SND-460V, SND-560, SNP-1000A, SNP-3300A, SNP-3350, SNP-3750, SNS-100, SNS-400, SNT-1010 - DVR : SHR-504X, SHR-604X, SVR-1670, SVR-1640A, SVR-1650E, SVR-950E, SVR-450, SVR-470, SVR-480, SRD-480D, SVR-940, SVR-945, SVR-960, SVR-960C, SVR-1645, SVR-1660, SVR-1660C, SVR-1680, SVR-1680C, SVR-3200 - NVR : SNR-3200, SNR-6400, SRN-3250, SRN-6450 - NET-i ware :SNS-SF064, SNS-SF032, SNS-SF016, SNS-SF008, SNS-SF004

nAddressType

[in] 주소 타입.

- XADDRESS_IP (=1): Static IP
- XADDRESS_URL (=2): URL
- XADDRESS_SSDDNS (=3): Samsung DDNS (MAC address를 사용함)
- XADDRESS_S1DDNS (=4): S1 DDNS (MAC address를 사용함)
- XADDRESS_IPOLISDDNS (=5): IPolis DDNS (ID를 사용함)

szAddress

[in] *nAddressType*에 따른 실제 주소. *nAddressType*에 따라 최대 길이가 달라짐. (표 2.2 참조)

nPort

[in] 포트 번호.

nHttpPort

[in] 웹(web) 접속을 위한 포트 번호.

szUserID

[in] 로그인용 ID. (최대 길이는 62바이트)

szPassword

[in] 로그인용 비밀번호. (최대 길이는 18바이트)

*szAddress*의 값은 *nAddressType*에 따라 아래와 같이 달라집니다.

표 2.2 주소 타입에 따른 실제 주소값

주소 타입(<i>nAddressType</i>)	Data Type	최대 길이(byte)	설명 및 예제
XADDRESS_IP	String	15	IPv4 IP address, "192.168.1.200"
XADDRESS_URL	String	254	URL, "habcdef.websamsung.net"
XADDRESS_SSDDNS	String	12	MAC address
XADDRESS_S1DDNS	String	12	MAC address
XADDRESS_IPOLISDDNS	String	20	'www.samsungipolis.com'에 등록된 ID, "mymodelid" (사용자가 임의로 생성함.)

Example

[C++]

```
long m_nDevice;
m_hDevice = m_ctlXnsDevice.CreateDevice(nDeviceID);

m_ctlXnsDevice.SetConnectionInfo(m_hDevice, "Samsung", "SHR-6164", XADDRESS_IP,
"192.168.10.94", 554, 80, "admin", "4321");
```

See Also

[CreateDevice](#)

SetConnectionInfo2

Description

이 함수는 SetConnectionInfo()와 같은 역할을 하며 SUNAPI 프로토콜 사용시 필요합니다. 멀티캐스트 사용시, 웹뷰어에서 멀티캐스트 관련 설정이 선행되어야 합니다.

Syntax

```
void SetConnectionInfo2(
    long hDevice,
    BSTR szVendorName,
    BSTR szmodelName,
    long nAddressType,
    BSTR szAddress,
    long nPort,
    long nHttpPort,
    BSTR szUserID,
    BSTR szPassword,
    long nDeviceProtocolType,
    long nMediaProtocolType
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

szVendorName

[in] 'Samsung'으로 고정임. (최대 길이는 126 바이트)

szmodelName

[in] 접속할 장비의 통합 모델명. (최대 길이는 126바이트. 표 2.2 참조)

nAddressType

[in] 주소 타입.

- XADDRESS_IP (=1): Static IP
- XADDRESS_URL (=2): URL
- XADDRESS_SSDDNS (=3): Samsung DDNS (MAC address를 사용함)
- XADDRESS_S1DDNS (=4): S1 DDNS (MAC address를 사용함)
- XADDRESS_IPOLISDDNS (=5): IPolis DDNS (ID를 사용함)

szAddress

[in] *nAddressType*에 따른 실제 주소. *nAddressType*에 따라 최대 길이가 달라짐. (오류! 참조 원본을 찾을 수 없습니다. 참조)

nPort

[in] 포트 번호.

nHttpPort

[in] 웹(web) 접속을 위한 포트 번호.

szUserID

[in] 로그인용 ID. (최대 길이는 62바이트)

szPassword

[in] 로그인용 비밀번호. (최대 길이는 18바이트)

nDeviceProtocolType

[in] 장비 프로토콜 타입

- XDEVICE_PROTOCOL_UNKNOWN (=0): 프로토콜 미지정

- XDEVICE_PROTOCOL_SUNAPI (=1): SUNAPI 프로토콜을 사용하도록 접속정보 설정.

nMediaProtocolType

[in] 미디어 프로토콜 타입

- XMEDIA_PROTOCOL_TCP (=0): TCP

- XMEDIA_PROTOCOL_UDP (=1): UDP

- XMEDIA_PROTOCOL_MULTICAST (=2): Multicast.. 라이브 모드에서만 유효

*szAddress*의 값은 *nAddressType*에 따라 달라집니다. (SetConnectionInfo()의 표 2.4 참조)

Example

[C++]

```
long m_nDevice;
m_hDevice = m_ctlXnsDevice.CreateDevice(nDeviceID);

m_ctlXnsDevice.SetConnectionInfo2(m_hDevice, "Samsung", "SNB-6004", XADDRESS_IP,
"192.168.10.94", 554, 80, "admin", "4321", XDEVICE_PROTOCOL_UNKNOWN, 0);
```

See Also

[CreateDevice](#)

GetDeviceID

Description

주어진 장비(device handle)의 device ID를 반환합니다.

Syntax

```
long GetDeviceID(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

Device ID. 리턴된 ID는 CreateDevice()의 파라미터로 사용됩니다. *hDevice* 값이 유효하지 않은 경우, -1(ERR_NOT_EXIST_DEVICE)이 리턴됩니다.

Example

[C++]

```
long nDeviceID = m_ctlXnsDevice.GetDeviceID(m_hDevice);
```

See Also

[CreateDevice](#), [GetDeviceHandle](#)

GetDeviceStatus

Description

장비의 접속 상태를 반환하는 함수입니다.

Syntax

```
long GetDeviceStatus(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- XDEVICE_STATUS_DISCONNECTED(0): 접속 해제 상태.
- XDEVICE_STATUS_CONNECTED(1): 접속 상태.
- ERR_NOT_EXIST_DEVICE(-1): *hDevice*가 유효하지 않음.

Example

[C++]

```
long nDeviceStatus = m_ctlXnsDevice.GetDeviceStatus(m_hDevice);
```

See Also[SetConnectionInfo](#)

GetVendorName

Description

장비의 벤더 이름을 반환하는 함수입니다.

Syntax

```
BSTR GetVendorName(long hDevice);
```

Parameters*hDevice*

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

벤더 이름. *hDevice*가 유효하지 않은 핸들인 경우, ""이 리턴됩니다.

Example**[C++]**

```
CString strVendor = m_ctlXnsDevice.GetVendorName(m_hDevice);
```

See Also[SetConnectionInfo](#)

GetModelName

Description

장비의 모델명을 반환하는 함수입니다.

Syntax

```
BSTR GetModelName(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

장비의 모델명. *hDevice*가 유효하지 않은 핸들인 경우, ""이 리턴됩니다.

Example

[C++]

```
CString strModel = m_ctlXnsDevice.GetModelName(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetVersion

Description

장비의 펌웨어(firmware) 버전을 반환하는 함수입니다. 이 함수는 애플리케이션이 장비에 접속한 이후에 사용할 수 있습니다.

Syntax

```
BSTR GetVersion(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

장비의 펌웨어 버전.

Example

[C++]

```
CString strVersion = m_ctlXnsDevice.GetVersion(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetAddressType

Description

장비에 설정된 주소 타입을 반환하는 함수입니다.

Syntax

```
long GetAddressType(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- XADDRESS_IP(1): Static IP
- XADDRESS_URL(2): URL
- XADDRESS_SSDDNS(3): Samsung DDNS
- XADDRESS_S1DDNS(4): S1 DDNS
- XADDRESS_IPOLISDDNS(5): IPolis DDNS

Example

[C++]

```
long addressType = m_ctlXnsDevice.GetAddressType(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetIP

Description

장비에 설정된 IP 주소를 반환합니다.

Syntax

```
BSTR GetIP(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

장비의 IP 주소. 장비가 동적 IP (dynamic IP)를 사용하는 경우, 이 값은 애플리케이션이 장비에 접속한 후, 실제 할당된 IP 주소로 채워집니다. 장비와 접속되어 있지 않은 경우에는 ""를 반환합니다.

Example

[C++]

```
CString strIp = m_ctlXnsDevice.GetIP(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetMac

Description

장비에 설정된 MAC address를 반환합니다.

Syntax

```
BSTR GetMac(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

MAC address. 애플리케이션이 SetConnectionInfo()를 이용하여 이 값을 설정하지 않은 경우, 이 값을 접속 후 채워집니다.

Example

[C++]

```
CString strMac = m_ctlXnsDevice.GetMac(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetURL

Description

장비에 설정된 URL을 반환합니다. 반환된 값은 주소 타입(*nAddressType*)이 XADDRESS_URL(=2)인 경우에만 유효합니다.

Syntax

```
BSTR GetURL(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

URL 주소.

Example

[C++]

```
CString strURL = m_ctlXnsDevice.GetURL(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetPort

Description

장비에 설정된 포트 번호를 반환합니다.

Syntax

```
long GetPort(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

장비의 포트 번호. 이 포트는 애플리케이션이 삼성의 프로토콜(SSNP, SVNP, VNP 및 DVR 프로토콜)을 이용하여 장비에 접속할 때 사용합니다.

Example

[C++]

```
long nPort = m_ctlXnsDevice.GetPort(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetHttpPort

Description

장비에 설정된 HTTP 포트 번호를 반환합니다. 이 포트 번호는 애플리케이션이 웹 브라우저를 이용하여 장비에 접속할 때 사용합니다.

Syntax

```
long GetHttpPort(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

장비의 HTTP 포트 번호.

Example

[C++]

```
long nHttpPort = m_ctlXnsDevice.GetHttpPort(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetUserID

Description

장비에 설정된 사용자 ID를 반환합니다. 이 값은 SetConnectionInfo()에 의해서 설정된 값과 동일합니다.

Syntax

```
BSTR GetUserID(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

사용자 ID. 이 값은 장비에 로그인 할 때 사용됩니다.

Example

[C++]

```
CString strUserId = m_ctlXnsDevice.GetUserID(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetPassword

Description

장비에 설정된 비밀번호를 반환합니다. 이 값은 SetConnectionInfo()에 의해서 설정된 값과 동일합니다.

Syntax

```
BSTR GetPassword(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

비밀 번호. 이 값은 장비에 로그인 할 때 사용됩니다.

Example

[C++]

```
CString strPassword = m_ctlXnsDevice.GetPassword(m_hDevice);
```

See Also

[SetConnectionInfo](#)

GetLatestUpdateTime

Description

장비의 마지막 업데이트 시간을 반환합니다. 최소 1회 이상 장비의 펌웨어를 업데이트 한 후에 이 함수를 호출해야 유효한 값을 얻을 수 있습니다. 업데이트 시간은 local PC의 장비 설정 파일에 기록됩니다.

Syntax

```
long GetLatestUpdateTime(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

마지막 업데이트 시간. 이 값의 자료형은 4바이트의 time_t 입니다.

Example

[C++]

```
time_t tLatestUpdateTime = m_ctlXnsDevice.GetLatestUpdateTime(m_hDevice);
```

GetDeviceHandle

Description

주어진 device ID에 대응하는 장비의 핸들(device handle)을 반환합니다. 이벤트가 발생하면 해당 이벤트 핸들러의 파라미터로 device ID가 전달됩니다. 그러나, XnsSdkDevice의 대부분의 함

수들은 장비 핸들을 파라미터로 요구하기 때문에, XnsSdkDevice의 함수들을 사용하려면, 이 함수를 이용하여 device ID를 장비 핸들로 변환하도록 합니다.

Syntax

```
long GetDeviceHandle(long nDeviceID);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

*nDeviceID*에 대응하는 장비의 핸들. *nDeviceID*에 대응하는 장비가 없는 경우, 0이 리턴됩니다.

Example

[C++]

```
long hDevice = m_tIXnsDevice.GetDeviceHandle(nDeviceID);
```

See Also

[GetDeviceID](#)

GetDeviceUTC

Description

DVR/NVR의 UTC Time을 반환합니다.

Syntax

```
long GetDeviceUTC(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

DVR/NVR의 UTC Time.

Example

[C++]

```
time_t tDeviceUTCTime = m_ctlXnsDevice.GetDeviceUTC(m_hDevice);
```

GetDeviceLocalTime

Description

DVR/NVR의 Local Time을 반환합니다.

Syntax

```
long GetDeviceLocalTime(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

DVR/NVR의 Local Time.

Example

[C++]

```
time_t tDeviceLocalTime = m_ctlXnsDevice.GetDeviceLocalTime(m_hDevice);
```

ChangeDeviceAdminPassword

Description

이 함수는 카메라 장비의 Admin계정 비밀번호를 변경하는 함수입니다. SUNAPI 프로토콜로 접속한 경우에만 가능합니다. 이 함수의 결과는 OnChangeDeviceAdminPassword 이벤트를 통해서 받을 수 있습니다.

Syntax

```
void ChangeDeviceAdminPassword(
    long hDevice,
    LPCTSTR szCurrentPassword,
    LPCTSTR szNewPassword,
    LPCTSTR szConfirmPassword
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

szCurrentPassword

[in] 현재 사용중인 비밀번호. 틀린 경우 ERR_PASSWORD_WRONG 반환

szNewPassword

[in] 새로 설정할 비밀번호.

szConfirmPassword

[in] 새로 설정할 비밀번호 확인. szNewPassword와 불일치할 경우

ERR_PASSWORD_NOT_MATCH 반환

Example

[C++]

```
m_ctlXnsDevice.ChangeDeviceAdminPassword(m_hDevice, m_strCurrentPw, m_strNewPw,
m_strConfirmPw);
```

See Also

[OnChangeDeviceAdminPassword](#)

Connect/Disconnect

Connect

Description

애플리케이션이 접속 정보를 이용하여 장비에 로그인하도록 하는 함수입니다. 접속 정보는 SetConnectionInfo()를 이용하여 설정해야 하며, 이 함수는 blocking function 방식으로 동작합니다. 네트워크 상태가 불안정한 경우, 이 함수는 최대 30초가 소요됩니다.

Syntax

```
long Connect(
    long hDevice,
    BOOL bForced,
```

```
    BOOL bReconnect
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

bForced

[in] 강제로 로그인 할지 결정하는 플래그. (DVR/NVR 제품군에서만 지원)

bReconnect

[in] 장비의 접속이 해제되면 자동 연결할 것인지를 결정하는 플래그. 이 값이 1인 경우, 재접속을 시도함.

참고

몇몇 장비는 admin 계정으로 동시에 여러 명이 접속하는 것을 허용하지 않습니다. 이 경우, 한 애플리케이션이 이미 접속 중이라면, 다른 애플리케이션들은 이 장비에 접속할 수 없습니다. 만약 *bForced* 파라미터가 1이라면, 이 애플리케이션은 장비에 접속할 수 있고, 이전에 접속하고 있던 애플리케이션은 강제로 접속이 종료됩니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.
- ERR_NOT_SUPPORTED_DEVICE: 지원하지 않는 장비임.
- ERR_EXCEED_CONNECTION: 연결 가능한 수를 초과함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_MAC_CONFLICT: MAC address가 충돌이 남.
- ERR_INVALID_VERSION: 유효하지 않은 버전임.
- ERR_INVALID_MODEL: 유효하지 않은 모델임.
- ERR_ALREADY_CONNECT: 장비와 이미 연결되어 있음.

Remarks

- 하나의 XNS ActiveX Device Control을 이용하여 같은 장비를 접속하려는 경우, [CreateDevice](#) 를 통해 별도의 디바이스 핸들러를 사용해야 합니다.

참고

한 프로세스 내 하나의 XNS ActiveX Device Control만 사용할 수 있습니다. 따라서 [Initialize](#) 작업 또한 한 번만 수행해야 합니다.

bForced 옵션은 현재 DVR/NVR 제품군에서만 동작합니다.

Example

[C++]

```
long Res = m_ctlXnsDevice.Connect(m_hDevice, 0, 0);

if (ERR_ADMIN_CONFLICT == Res)
    Res = m_ctlXnsDevice.Connect(m_hDevice, 1, 0);
```

See Also

[ConnectNonBlock](#)

ConnectNonBlock

Description

애플리케이션이 접속 정보를 이용하여 장비에 로그인하도록 하는 함수입니다. 접속 정보는 SetConnectionInfo()를 이용하여 설정합니다. 이 함수는 non-blocking function 방식으로 동작하며, 접속이 완료되지 않아도 바로 리턴됩니다. 접속 결과는 이벤트를 통해 전달되며, 접속 성공 시 OnDeviceStatusChanged 이벤트가 발생합니다. 접속에 실패하면, OnConnectFailed 이벤트가 발생합니다.

Syntax

```
long ConnectNonBlock(
    long hDevice,
    BOOL bForced,
    BOOL bReconnect
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

bForced

[in] 강제로 로그인 할지 결정하는 플래그. (DVR/NVR 제품군만 지원)

bReconnect

[in] 장비의 접속이 해제되면 자동 연결할 것인지를 결정하는 플래그. 이 값이 1인 경우, 재접속을 시도함.

참고

몇몇 장비는 admin 계정으로 동시에 여러 명이 접속하는 것을 허용하지 않습니다. 이 경우, 한 애플리케이션이 이미 접속 중이라면, 다른 애플리케이션들은 이 장비에 접속할 수 없습니다.

만약 *bForced* 파라미터가 1이라면, 이 애플리케이션은 장비에 접속할 수 있고, 이전에 접속하고 있던 애플리케이션은 강제로 접속이 종료됩니다.

Remarks

- 하나의 XNS ActiveX Device Control을 이용하여 같은 장비를 접속하려는 경우, [CreateDevice](#)를 통해 별도의 디바이스 핸들러를 사용해야 합니다.

참고

한 프로세스 내 하나의 XNS ActiveX Device Control만 사용할 수 있습니다. 따라서 [Initialize](#) 작업 또한 한 번만 수행해야 합니다.

bForced 옵션은 현재 DVR/NVR 제품군에서만 동작합니다.

Example

[C++]

```
long Res = m_ctlXnsDevice.ConnectNonBlock(m_hDevice, 0, 0);
```

See Also

[Connect](#), [OnConnectFailed](#), [OnDeviceStatusChanged](#), [SetConnectionInfo](#)

Disconnect

Description

장비와의 접속을 해제합니다.

Syntax

```
long Disconnect(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long Res = m_ctlXnsDevice.Disconnect(m_hDevice);
```

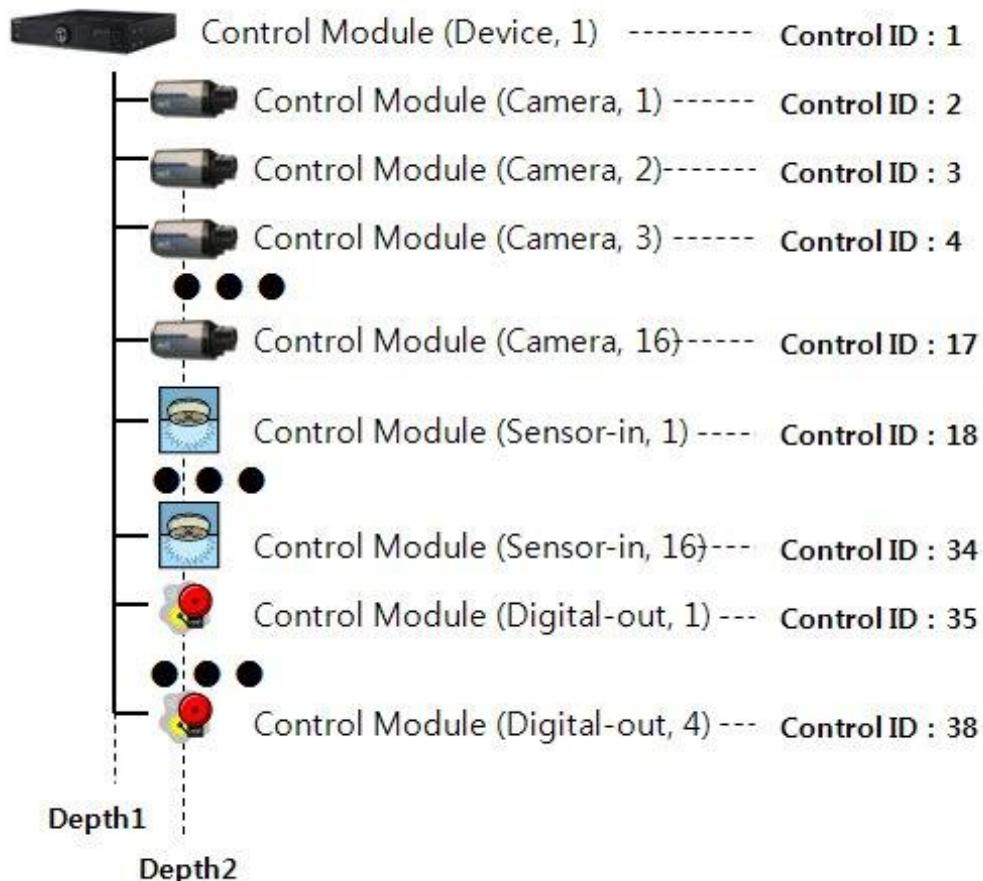
See Also

[Connect](#), [ConnectNonBlock](#)

Control Module Information

하나의 제품은 여러 개의 독립된 control module로 구성되어 있습니다. 각 control module은 기능(capabilities)을 갖고 있습니다. Control module은 아래 그림과 같이 트리 구조를 가지며 각 모듈에는 control ID가 부여됩니다.

그림 2.1 DVR(SRD-1670)의 Control module 구조



위의 그림에서 보는 바와 같이, DVR 자체도 하나의 control module입니다. 하나의 DVR에는 16 개의 카메라가 연결되어 있고 각 카메라는 control module입니다. 애플리케이션은 각 카메라에 서 스트림 데이터 또는 저장된 데이터를 가져올 수 있습니다. 위의 그림처럼 여러 대의 카메라가 연결되어 있는 경우, 이를 구분하기 위해 인덱스(index)를 사용하는데, 이를 control 번호라고 부릅니다.

예를 들어, 위의 그림에서 첫 번째 카메라의 경우 Control ID는 2, Control 번호는 1입니다.

이 DVR은 카메라 외에 sensor-input과 alarm-out 포트를 갖고 있으며 이 포트들은 on/off 상태 정보를 가질 수 있고, 애플리케이션은 alarm-out 포트를 제어할 수 있습니다. 따라서, 이 포트 역시 control ID를 가집니다.

그림 2.2 멀티프로파일 지원하는 DVR/NVR Control module 구조

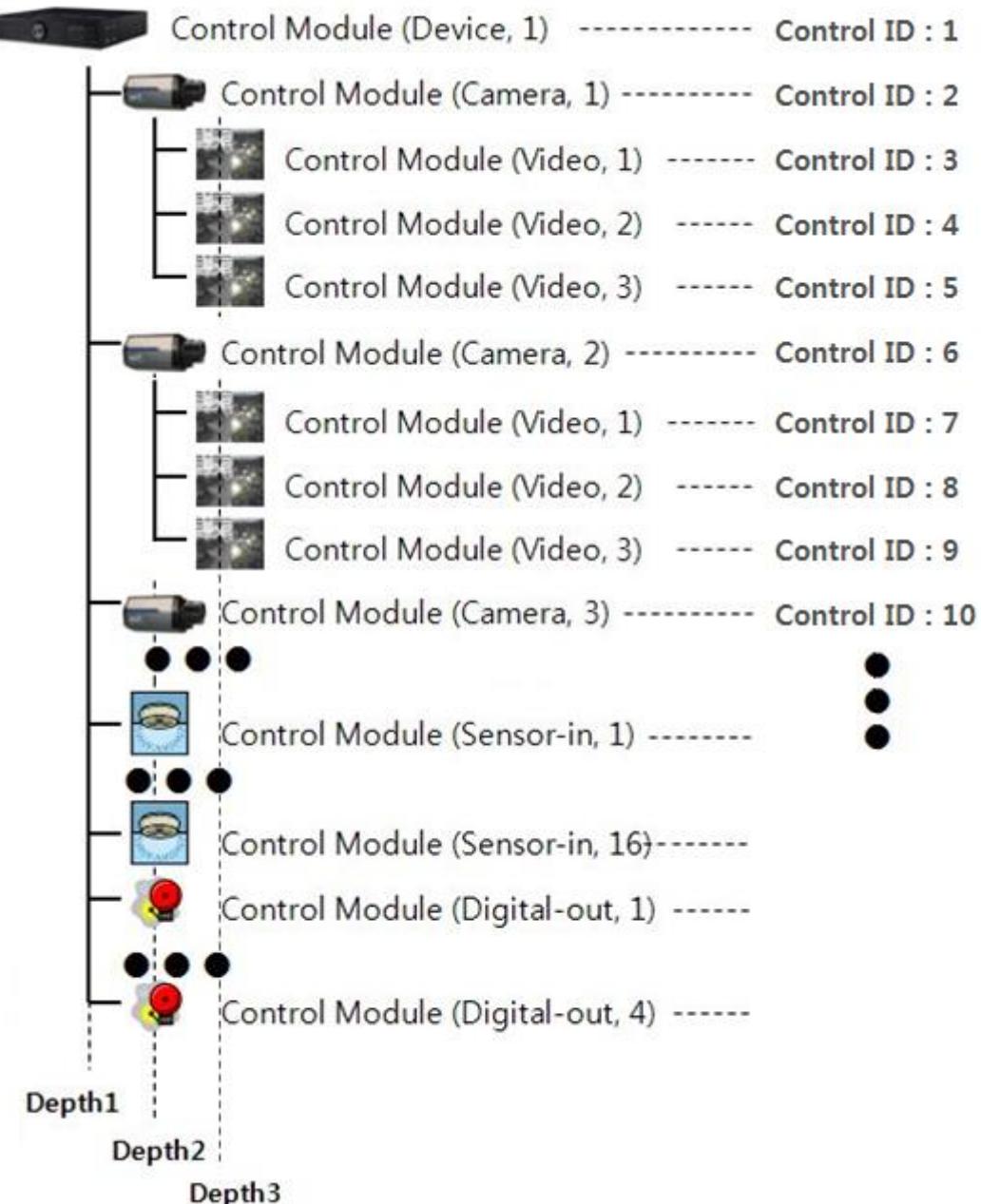
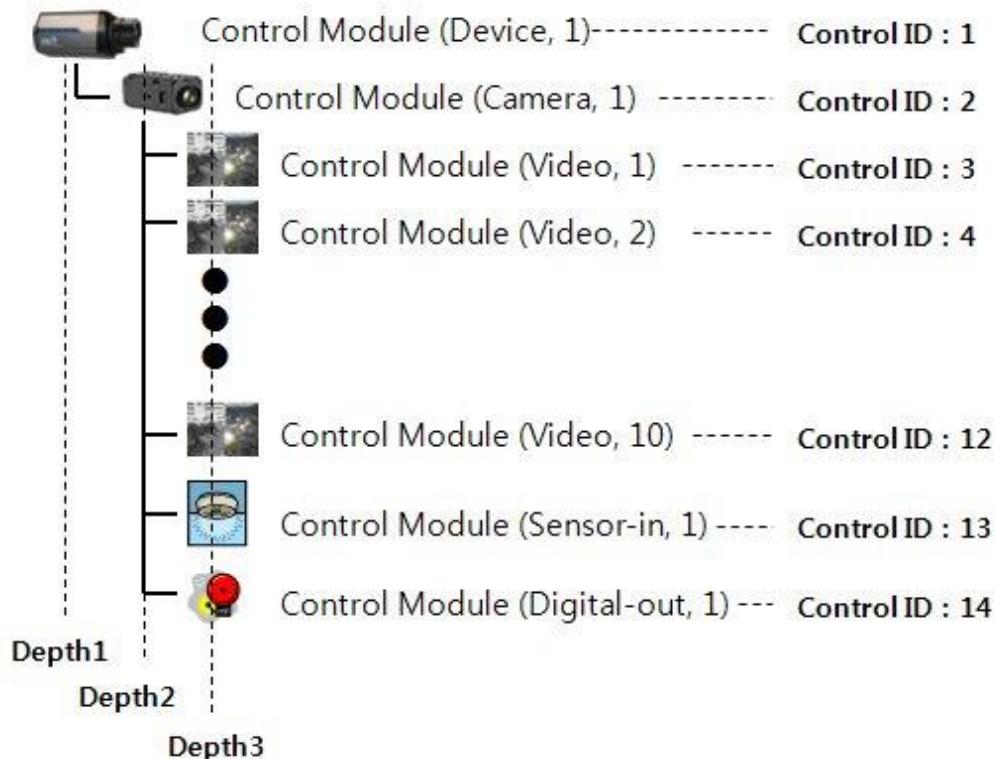


그림 2.3 네트워크 카메라(SNB-5000)의 Control module 구조



네트워크 카메라는 multiple stream을 지원합니다. 이런 경우, 애플리케이션은 미리 정의된 스트림 포맷 (즉, profile)을 사용하여 스트림을 가져옵니다. 따라서, 네트워크 카메라는 3-depth의 트리 구조를 가집니다.

각 control module은 수행할 수 있는 기능(capability)을 가지는데, 이는 모듈 타입에 따라 다릅니다. 각 모듈 타입에 따른 capability는 아래 표를 참조하시기 바랍니다.

표 2.3 Control module type에 따른 capability

Type	Value	Description
XCTL_ALL	0xFFFF	Support all types.
XCTL_DVR	0x0001	DVR.
XCTL_NETCAM	0x0002	Network Camera.
XCTL_ENCODER	0x0004	Encoder Box (Video server)
XCTL_CAMERA	0x0008	Camera channel of DVR.
XCTL_VIDEO	0x0010	Possible video stream of network camera.
XCTL_ALARM	0x0020	Alarm output device.
XCTL_SENSOR	0x0040	Sensor input device. (Digital input, alarm input)
XCTL_TEXT	0x0800	Text device. (POS)

| 참고

장비 자신(즉, 최상위 장비)은 항상 Control ID가 1이어야 합니다.

GetControlCount

Description

Control module의 수를 반환합니다. 애플리케이션은 특정 타입의 control module의 수를 얻을 수 있으며, 비디오 또는 카메라의 전체 개수를 구할 수도 있습니다.

Syntax

```
long GetControlCount(
    long hDevice,
    long nType
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nType

[in] Control module의 타입. ([표 2.3참조](#))

Return Value

*nType*의 Control module 수.

Example

[C++]

```
long m_nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);
```

See Also

[GetControlType](#)

GetControlType

Description

Control ID에 대응하는 control module의 타입을 반환합니다.

Syntax

```
long GetControlType(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

Control module 타입. 표 2.3 Control module type에 따른 capability 참조.

Example

[C++]

```
long nType;
long m_nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);

for (int i=0; i<m_nControlCount; i++)
{
    nType = m_ctlXnsDevice.GetControlType(m_hDevice, i);
}
```

See Also

[GetControlCount](#), [ControlID](#)

GetControlNumber

Description

Control ID에 대응하는 control module의 번호를 반환합니다.

Syntax

```
long GetControlNumber(
    long hDevice,
    long nControlID
```

);

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

Control module 번호.

Example

[C++]

```
long m_nNumber;
long m_nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);

for (int i=0; i<m_nControlCount; i++)
{
    m_nNumber = m_ctlXnsDevice.GetControlNumber(m_hDevice, i);
}
```

See Also

[GetControlCount](#), [ControlID](#)

GetControlDepth

Description

Control ID에 대응하는 control module의 depth를 반환합니다.

Syntax

```
long GetControlDepth(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

Control module의 depth.

Example

[C++]

```
long m_nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);

for (int i=0; i<m_nControlCount; i++)
{
    m_nDepth[i] = m_ctlXnsDevice.GetControlDepth(m_hDevice, i);
}
```

See Also

[GetControlCount](#), [ControlID](#)

GetControlLocalName

Description

애플리케이션에 저장된 control module의 이름을 반환합니다. 초기 이름의 포맷은 "IpAddress:portNumber-channelNumber"입니다. 이 이름을 변경하려면 ChangeControlLocalName()을 사용합니다.

Syntax

```
BSTR GetControlLocalName(
    long hDevice,
    long nControlID
);
```

Parameters*hDevice*

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

Control module 이름.

Example

[C++]

```
void CTestSampleDlg::OnControlLocalNameChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID)
{
    OutputDebugString("Event Local Name Changed\n");
    char str[300];
    int hDevice = m_ctlXnsDevice.GetDeviceHandle(nDeviceID);
    sprintf(str, "nDeviceID[%d], nControlID[%d], Changed Name[%s]\n", nDeviceID,
nControlID, m_ctlXnsDevice.GetControlLocalName(hDevice, nControlID));
    OutputDebugString(str);
}
```

See Also

[OnControlLocalNameChanged](#), [ChangeControlLocalName](#), [ControlID](#)

GetControlRemoteName

Description

장비에 저장된 control module의 이름을 반환합니다.

Syntax

```
BSTR GetControlRemoteName(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

장비에 저장된 control module의 이름. 반환값이 가질 수 있는 최대 길이는 127바이트입니다.

Example

[C++]

```
long m_nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);

CString strRemoteName[m_nControlCount];
for (int i=0; i<m_nControlCount; i++)
{
    strName[i] = m_ctlXnsDevice.GetControlRemoteName(m_hDevice, i);
}
```

See Also

[GetControlLocalName](#), [OnControlRemoteNameChanged](#), [ControlID](#)

GetControlStatus

Description

Control module의 상태를 가져오는 함수입니다.

Syntax

```
long GetControlStatus(
    long hDevice,
    long nControlID,
    long nStatusID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nStatusID

[in] 상태 정보의 각 항목을 나타내는 ID. (표 2.4 참조)

표 2.4 Status ID

Name	ID	Description
XCAMERA_STATUS	1	카메라의 on/off 상태를 나타냄.
XRECORDING_STATUS	2	장비의 녹화(recording) 상태를 나타냄.
XLISTEN_STATUS	3	Listen 기능의 동작 상태를 나타냄.
XTALK_STATUS	4	Talk 기능의 동작 상태를 나타냄.
XALARM_STATUS	5	Alarm-out 포트의 상태를 나타냄.
XLOCALREC_STATUS	6	Local 장비의 녹화 상태를 나타냄.

Return Value

각 *nStatusID*에 대해서 이 함수가 반환하는 값은 다음과 같은 의미를 가집니다.

표 2.5 GetControlStatus()의 Return Value

Status ID	Return Value
XCAMERA_STATUS	XCAM_STATUS_ON (=0x00000001)
	XCAM_STATUS_SIGNAL (=0x00000002)
	XCAM_STATUS_COVERT1 (=0x00000004)
	XCAM_STATUS_COVERT2 (=0x00000008)
	XCAM_STATUS_ACTIVATED (=0x00000010)
XRECORDING_STATUS	0
	1
XLISTEN_STATUS	0
	1
XTALK_STATUS	0
	1
XALARM_STATUS	0
	1
XLOCALREC_STATUS	0
	1

Status ID 가 XCAMERA_STATUS일 경우 반환 값은 bit-wise ORing 연산으로 조합될 수 있습니다. 예를 들면 19(0x13), 17(0x11), 21(0x15)은 아래와 같은 의미를 가지고 있습니다.

Returned value	Bit-wise ORing	Meaning
16(0x10)	XCAM_STATUS_ACTIVATED	Connect, Video Off
17(0x11)	XCAM_STATUS_ON XCAM_STATUS_ACTIVATED	Connect, Video On, Media On / Connect, VideoLoss
19(0x13)	XCAM_STATUS_ON XCAM_STATUS_ACTIVATED XCAM_STATUS_SIGNAL	Connect, Video On, Media On / Connect
21(0x15)	XCAM_STATUS_ON XCAM_STATUS_COVERT1 XCAM_STATUS_ACTIVATED	Connect, Covert

Example

[C++]

```
long m_nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);

for (int i=0; i<m_nControlCount; i++)
{
    nStatus = m_ctlXnsDevice.GetControlStatus(m_hDevice, i, XCAMERA_STATUS);
}
```

See Also

[OnControlStatusChanged](#), [ControlID](#)

GetControlCapability

Description

Control module의 기능(capability)를 반환합니다.

Syntax

```
BOOL GetControlCapability()
```

```

        long hDevice,
        long nControlID,
        long nCapabilityID
);

```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nCapabilityID

[in] Capability ID. (표 2.6 참조)

표 2.6 Capability ID

Field	Capability ID	Description
XCTL_CAP_NONE	0	사용하지 않음.
XCTL_CAP_SW_UPLOAD	1	장비의 펌웨어를 원격으로 업데이트함.
XCTL_CAP_CFG_IMPORT	2	장비의 설정을 가져오는 기능.
XCTL_CAP_CFG_EXPORT	3	장비의 설정을 내보내는 기능.
XCTL_CAP_CONN_FORCED	4	관리자 계정으로 강제 접속할 수 있는 기능.
XCTL_CAP_MEDIA_AUTHORIT Y	5	미디어 권한을 가져오는 기능.
XCTL_CAP_FSPEED1	6	1 배속 재생.
XCTL_CAP_FSPEED2	7	2 배속 재생.
XCTL_CAP_FSPEED4	8	4 배속 재생.
XCTL_CAP_FSPEED8	9	8 배속 재생.
XCTL_CAP_FSPEED16	10	16 배속 재생.
XCTL_CAP_FSPEED32	11	32 배속 재생.
XCTL_CAP_FSPEED64	12	64 배속 재생.
XCTL_CAP_BSPEED1	13	1 배속 역재생.
XCTL_CAP_BSPEED2	14	2 배속 역재생.
XCTL_CAP_BSPEED4	15	4 배속 역재생.
XCTL_CAP_BSPEED8	16	8 배속 역재생.

XCTL_CAP_BSPEED16	17	16 배속 재생.
XCTL_CAP_BSPEED32	18	32 배속 재생.
XCTL_CAP_BSPEED64	19	64 배속 재생.
XCTL_CAP_FSSPEED1	20	1 배속 재생.
XCTL_CAP_FSSPEED2	21	1/2 배속 재생.
XCTL_CAP_FSSPEED4	22	1/4 배속 재생.
XCTL_CAP_FSSPEED8	23	1/8 배속 재생.
XCTL_CAP_FSSPEED16	24	1/16 배속 재생.
XCTL_CAP_FSSPEED32	25	1/32 배속 재생.
XCTL_CAP_FSSPEED64	26	1/64 배속 재생.
XCTL_CAP_BSSPEED1	27	1 배속 재생.
XCTL_CAP_BSSPEED2	28	1/2 배속 재생.
XCTL_CAP_BSSPEED4	29	1/4 배속 재생.
XCTL_CAP_BSSPEED8	30	1/8 배속 재생.
XCTL_CAP_BSSPEED16	31	1/16 배속 재생.
XCTL_CAP_BSSPEED32	32	1/32 배속 재생.
XCTL_CAP_BSSPEED64	33	1/64 배속 재생.
XCTL_CAP_SEEK	34	탐색 기능.
XCTL_CAP_FSTEP	35	앞으로 감기.
XCTL_CAP_BSTEP	36	뒤로 감기.
XCTL_CAP_LIVE	37	실시간 영상 출력.
XCTL_CAP_PLAYBACK	38	영상 재생.
XCTL_CAP_BACKUP	39	영상 백업.
XCTL_CAP_SEARCH	40	영상 검색.
XCTL_CAP_LISTEN	41	Listen 기능.
XCTL_CAP_TALK	42	Talk 기능.
XCTL_CAP_CAM_MENU	43	화면 상에서 카메라 메뉴를 제어.
XCTL_CAP_PTZ_PAN	44	Pan 이동 기능.
XCTL_CAP_PTZ_TILT	45	Tilt 이동 기능.
XCTL_CAP_PTZ_ZOOM	46	줌(zoom) 조정 기능.
XCTL_CAP_PTZ_SPEED	47	PTZ 를 속도에 따라 제어하는 기능.

XCTL_CAP_PTZ_PRESET	48	프리셋 설정 기능.(HPTZ)
XCTL_CAP_PTZ_PRESET_SVR	49	프리셋 설정 기능 (SVR 모델)
XCTL_CAP_PTZ_AUTOPAN	50	오토팬 기능
XCTL_CAP_PTZ_SCAN	51	스캔 기능
XCTL_CAP_PTZ_PATTERN	52	패턴 기능
XCTL_CAP_PTZ_SWING	53	스윙 기능
XCTL_CAP_PTZ TOUR	54	투어 기능
XCTL_CAP_PTZ_POWER	55	Power PTZ (PPTZ) 기능.
XCTL_CAP_PTZ_FREE_MOVE	56	PTZ 이동 기능.
XCTL_CAP_PTZ_SPEED_ZOOM	57	속도에 따라 PTZ 를 이동하는 기능.
XCTL_CAP_CAM_IRIS	58	Iris open/close 하는 기능.
XCTL_CAP_CAM_FOCUS	59	Focus in/out 기능.
XCTL_CAP_CAM_BRIGHTNESS	60	밝기 조절 기능.
XCTL_CAP_CAM_CONTRAST	61	대비 조절 기능.
XCTL_CAP_ALARM_ONOFF	62	Alarm Out 을 On/Off 하는 기능.
XCTL_CAP_DVR_EVENT_RESET	63	DVR 이벤트 리셋 기능
XCTL_CAP_DVR_MANUAL_RECORD_CONTROL	64	DVR 매뉴얼 레코드 기능
XCTL_CAP_DVR_REC_STATUS	65	DVR 레코드 상태 기능
XCTL_CAP_MOTION_SEARCH	66	모션 이벤트 검색 기능
XCTL_CAP_PTZ_AREA_ZOOM	67	Area Zoom 기능
XCTL_CAP_PTZ_GET_POS	68	PTZ 절대좌표값을 GET 하는 기능
XCTL_CAP_PTZ_SET_POS	69	PTZ 절대좌표값을 SET 하는 기능
XCTL_CAP_OVERLAP_TRACK_SEARCH	70	중복구간 검색 기능
XCTL_CAP_POS_SEARCH	71	POS 검색 기능
XCTL_CAP_PTZ_GET_POS_SPEED	72	PTZ 절대좌표 이동시 속도값 GET 기능
XCTL_CAP_PTZ_SET_POS_SPEED	73	PTZ 절대좌표 이동시 속도값 SET 기능

XCTL_CAP_POWER_RESTART	74	전원 재시작 기능
XCTL_CAP_BANDWIDTH	75	Bandwidth 설정 기능
XCTL_CAP_PTZ_GET_POS_NORMALIZE	86	PTZ 정규화 좌표 GET 하는 기능
XCTL_CAP_PTZ_SET_POS_NORMALIZE	87	PTZ 정규화 좌표 SET 하는 기능
XCTL_CAP_TEXT	89	Text Device 지원

Return Value

장비가 Capability ID에 해당하는 Capability를 제공하는지 여부를 반환합니다.

- TRUE: 장비가 해당 capability를 제공합니다.
- FALSE: 장비가 해당 capability를 제공하지 않습니다.

Example

[C++]

```
// 아래 함수는 장비의 상태가 변경되었을 때,
// 호출되는 OnDeviceStatusChanged 이벤트에 대한 이벤트 핸들러입니다.

void CMY01_SimpleLiveDlg::OnDeviceStatusChangedXnssdkdevicectrl1(long nDeviceID, long
nErrorCode, long nDeviceStatus, long nHddCondition)
{
    if (nErrorCode == ERR_SUCCESS && nDeviceStatus == 1)
    {
        for(int i=0; i<m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_CAMERA);
i++)
        {
            if (m_ctlXnsDevice.GetControlCapability(m_hDevice, 2,
XCTL_CAP_LIVE))
            {
                m_ctlXnsDevice.OpenMedia(m_hDevice, i+2,
MEDIA_TYPE_LIVE, 0, 0, &m_hMediaSource[i]);
                m_ctlXnsWindow[i].Start(m_hMediaSource[i]);
            }
        }
    }
}
```

See Also

[ControlID](#)

ChangeControlLocalName

Description

Control module의 local PC에 설정된 이름을 변경합니다.

Syntax

```
void ChangeControlLocalName(
    long hDevice,
    long nControlID,
    BSTR szName
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

szName

[in] 변경할 이름. 최대 길이는 127바이트임.

Remarks

Control module의 이름이 변경된 후에는, OnControlNameChanged 이벤트가 발생합니다.

Example

[C++]

```
CString strName = "Local Change Name";
m_ctlXnsDevice.ChangeControlLocalName(m_hDevice, 2, strName);
```

See Also

[OnControlLocalNameChanged](#), [ControlID](#)

GetControlLinkStatus

Description

Control module의 카메라 Link 상태를 가져오는 함수입니다.

Syntax

```
BSTR GetControlLinkStatus(
    long hDevice,
    long nControlID,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Text device control module의 control ID.

Return Value

Control module의 카메라 Link 상태를 반환합니다. 다음 Control type에 해당하는 Control ID만 유합니다.

표 2.7 GetControlLinkStatus()에서 유효한 Control Type

Type	Value	Description
XCTL_SENSOR	0x0040	Sensor input device. (Digital input, alarm input)
XCTL_TEXT	0x0800	Text device. (POS)

Example

[C++]

```
long nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);
CString strLinkStatus;
for (int i=1; i<nControlCount; i++)
{
    if( m_ctlXnsDevice.GetControlType(m_hDevice, i) == XCTL_TEXT )
    {
        strLinkStatus = m_ctlXnsDevice.GetControlLinkStatus(m_hDevice, i);
        break;
    }
}
```

See Also

[OnControlStatusChanged](#), [ControlID](#)

GetVideoControlID

Description

Camera Module과 Video Module 번호를 설정하면 해당 ControlID를 반환합니다.

Syntax

```
long GetVideoControlID(
    long hDevice,
    long CameraIndex,
    long VideoIndex
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nCameraIndex

[in] Camera Module의 번호.

nVideoIndex

[in] Video Module의 번호.

Return Value

설정한 Camera Module, Video Module의 ControlID.

Example

[C++]

```
long nControlID = m_ctlXnsDevice.GetVideoControlID(m_hDevice, m_nChannelIndex,
m_nProfileIndex);
```

See Also

[ControlID](#)

GetParentControlID

Description

상위 control module의 ID를 반환합니다.

Syntax

```
long GetParentControlID(  
    long hDevice,  
    long nControlID  
)
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

상위 control module의 ID.

Example

[C++]

```
long nParentControlID = m_ctlXnsDevice.GetParentControlID(m_hDevice, nControlID);
```

See Also

[ControlID](#)

GetVideoResolution

Description

장비에 저장된 control module의 Video Resolution을 가져옵니다.

Syntax

```
long GetVideoResolution(  
    long hDevice,  
    long nControlID,  
    long* pnVideoWidth,  
    long* pnVideoHeight)
```

);

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

pnVideoWidth

[out] 비디오 해상도 폭

pnVideoHeight

[out] 비디오 해상도 높이

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long nVideoWidth = 0;
long m_nVideoHeight = 0;
m_ctlXnsDevice.GetVideoResolution(m_hDevice, m_nControlID, &nVideoWidth, &nVideoHeight);
```

See Also

[ControlID](#)

GetVideoFrameRate

Description

장비에 저장된 control module의 Video Frame Rate를 반환합니다.

Syntax

```
long GetVideoFrameRate(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

장비에 저장된 control module의 Frame Rate.

Example

[\[C++\]](#)

```
long nFrameRate = 0;
nFrameRate = m_ctlXnsDevice.GetVideoFrameRate(m_hDevice, m_nControlID);
```

See Also

[ControlID](#)

GetVideoQuality

Description

장비에 저장된 control module의 Video Quality를 반환합니다.

Syntax

```
long GetVideoQuality(
    long hDevice,
    long nControlID
);
```

Parameters*hDevice*

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

장비에 저장된 control module의 Video Quality.

Example

[C++]

```
long nVideoQuality = 0;
nVideoQuality = m_ctlXnsDevice.GetVideoQuality(m_hDevice, m_nControlID);
```

See Also

[ControlID](#)

GetVideoCodec

Description

장비에 저장된 control module의 Video Codec을 반환합니다.

Syntax

```
BSTR GetVideoCodec(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

장비에 저장된 control module의 Video Codec 이름. 반환값이 가질 수 있는 최대 길이는 63바이트입니다.

- "JPEG" : JPEG
- "MJPEG" : MJPEG
- "MP4V" : MPEG4
- "H264" : H264
- "UNKNOWN" : Unknown

Example

[C++]

```
CString strVideoCodec;
strVideoCodec = m_ctlXnsDevice.GetVideoCodec(m_hDevice, m_nControlID);
```

See Also[ControlID](#)

Start/End to Play Media

AcquireMediaAuthority

Description

녹화된 미디어를 재생하기 위한 권한을 획득하는 함수입니다. 애플리케이션은 영상을 재생하기 전에 이 함수를 이용하여 반드시 권한을 얻어야 합니다.

Syntax

```
long AcquireMediaAuthority(long hDevice);
```

Parameters*hDevice*

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NOT_SUPPORTED_COMMAND: 지원하지 않는 명령임.
- ERR_NO_PERMISSION: 권한이 없음.

Example[\[C++\]](#)

```
long ret = m_ctlXnsDevice.AcquireMediaAuthority(m_hDevice);
```

See Also

[OpenMedia](#), [Play](#), [Pause](#), [Seek](#), [FrameAdvance](#), [SearchCalendar](#), [SearchDay](#)

ReleaseMediaAuthority

Description

미디어를 재생하기 위해 획득한 권한을 해제합니다. 애플리케이션은 미디어 재생 후 반드시 이 함수를 이용하여 재생 권한을 해제해야 합니다.

Syntax

```
long ReleaseMediaAuthority(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.ReleaseMediaAuthority(m_hDevice);
```

See Also

[OpenMedia](#), [Play](#), [Pause](#), [Seek](#), [FrameAdvance](#), [SearchCalendar](#), [SearchDay](#)

OpenMedia

Description

이 함수를 호출하면, 장비로부터 미디어 스트림을 수신하기 시작합니다.

수신된 미디어 스트림은 XnsSdkWindow 컴포넌트로 전달되며, XnsSdkWindow는 수신한 스트림을 디코딩(decoding)하여 재생합니다.

스트림 데이터를 XnsSdkWindow에 연결하기 위해서는 *phMediaSource*가 필요합니다. 이 값은

OpenMedia()의 파라미터(out-parameter)로 얻을 수 있으며, 이 값을 XnsSdkWindow에 전달하면 XnsSdkWindow는 장비로부터 스트림 데이터를 얻을 수 있습니다.

*phMediaSource*는 미디어 재생을 제어하는 데에도 사용됩니다. 따라서, 애플리케이션은 이 값을 잘 유지하고 있어야 합니다.

Syntax

```
long OpenMedia(
    long hDevice,
    long nControlID,
    long nMediaType,
    long tStart,
    long tEnd,
    long* phMediaSource
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nMediaType

[in] 미디어 타입

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

tStart

[in] 재생 시작 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

tEnd

[in] 재생 종료 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

phMediaSource

[out] Media stream ID. 이 값은 미디어 스트림을 제어하는데 필요하며, XnsSdkWindow::Start()의 파라미터로 사용됨.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.
- ERR_MEDIA_ALREADY_OPEN: 미디어가 이미 열려있음.

Example

[C++]

```
long m_pMediaLiveSource;
long ret;
ret = m_ctlXnsDevice.OpenMedia(m_hDevice, 2, XMEDIA_LIVE, 0, 0, &m_pMediaLiveSource);
```

See Also

[OpenMediaEx](#), [CloseMedia](#), [XnsSdkWindow::Start](#), [ControlID](#)

OpenMediaEx

Description

OpenMeida와 같은 역할을 하는 함수로 자바스크립트를 지원하는 함수입니다.

Syntax

```
long OpenMediaEx(
    long hDevice,
    long nControlID,
    long nMediaType,
    long tStart,
    long tEnd
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nMediaType

[in] 미디어 타입

- XMEDIA_LIVE (=1): Live

- XMEDIA_PLAYBACK (=2): Playback

- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

tStart

[in] 재생 시작 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

tEnd

[in] 재생 종료 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

Return Value

미디어 스트림 핸들값 리턴. 0일 경우 실패

Example

[C++]

```
long hMediaSource = 0;
hMediaSource = m_ctlXnsDevice.OpenMediaEx(m_hDevice, 2, XMEDIA_LIVE, 0, 0);
```

See Also

[OpenMedia](#), [CloseMedia](#), [XnsSdkWindow::Start](#), [ControlID](#)

CloseMedia

Description

장비로부터 미디어 스트림이 전송되는 것을 중단시킵니다. 미디어 스트림은 *hMediaSource*(즉, OpenMedia()의 *phMediaSource*)에 의해 구별됩니다.

Syntax

```
long CloseMedia(
    long hDevice,
    long hMediaSource
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

hMediaSource

[in] Media stream ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

Example

[C++]

```
long ret;
long m_pMediaLiveSource;
ret = m_ctlXnsDevice.OpenMedia(m_hDevice, 2, XMEDIA_LIVE, 0, 0, &m_pMediaLiveSource);

// Write your code
// ...

ret = m_ctlXnsDevice.CloseMedia(m_hDevice, m_pMediaLiveSource);
```

See Also

[OpenMedia](#), [OpenMediaEx](#)

OpenStream

Description

장비로부터 미디어 스트림 수신을 시작합니다. 수신된 미디어 스트림은 [OnNewMedia](#) 이벤트로 전달됩니다. 애플리케이션은 이 데이터를 디코딩(decoding)하거나 XnsSdkWindow에 전달하여 재생할 수 있습니다.

Syntax

```
long OpenStream(
    long hDevice,
    long nControlID,
    long nType,
    long tStart,
    long tEnd
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

tStart

[in] 재생 시작 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

tEnd

[in] 재생 종료 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.
- ERR_MEDIA_ALREADY_OPENED: 미디어가 이미 열려있음.

Example

[C++]

```
long ret = m_ctlXnsDevice.OpenStream(m_hDevice, 2, XMEDIA_LIVE, 0, 0);
```

See Also

[CloseStream](#), [XnsSdkWindow::Start](#), [ControlID](#), [OnNewMedia](#)

CloseStream

Description

미디어 스트림 전송을 중단합니다. 미디어 스트림은 Control ID를 이용하여 구분합니다.

Syntax

```
long CloseStream(
    long hDevice,
```

```
    long nControlID,
    long nType
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

Example

[C++]

```
long ret = m_ctlXnsDevice.CloseStream(m_hDevice, 2, XMEDIA_LIVE);
```

See Also

[OpenStream](#), [ControlID](#)

SuspendStream

Description

미디어 스트림 데이터의 전송을 일시 중지합니다. 이 스트림은 Control ID로 구분합니다.

Syntax

```
long SuspendStream(
    long hDevice,
    long nControlID,
```

```
    long nMediaType  
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nMediaType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

Example

[C++]

```
long ret = m_ctlXnsDevice.SuspendStream(m_hDevice, 2, XMEDIA_LIVE);
```

See Also

[ResumeStream](#), [SuspendAllStream](#), [ControlID](#)

SuspendAllStream

Description

모든 미디어 스트림 전송을 일시 중지합니다.

Syntax

```
long SuspendAllStream(  
    long hDevice,  
    long nMediaType  
)
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nMediaType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

Example

[C++]

```
long ret;
ret = m_ctlXnsDevice.SuspendAllStream(m_hDevice, XMEDIA_LIVE);
```

See Also

[SuspendStream](#), [ResumeAllStream](#)

ResumeStream

Description

일시 정지된 미디어 스트림의 전송을 재개합니다. 미디어 스트림은 Control ID로 구분됩니다.

Syntax

```
long ResumeStream(
    long hDevice,
    long nControlID,
    long nMediaType
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nMediaType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

Example

[C++]

```
long ret = m_ctlXnsDevice.ResumeStream(m_hDevice, 2, XMEDIA_LIVE);
```

See Also

[SuspendStream](#), [SuspendAllStream](#), [ResumeAllStream](#), [ControlID](#)

ResumeAllStream

Description

일시 정지된 모든 미디어 스트림의 전송을 재개합니다.

Syntax

```
long ResumeAllStream(
    long hDevice,
    long nMediaType
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nMediaType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

Return Value

ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long ret = m_ctlXnsDevice.ResumeAllStream(m_hDevice, XMEDIA_LIVE);
```

See Also

[SuspendStream](#), [SuspendAllStream](#), [ResumeStream](#)

IsPlaying

Description

미디어 스트림이 재생 중인지를 알려줍니다.

Syntax

```
BOOL IsPlaying(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- TRUE: 미디어가 재생 중입니다.
- FALSE: 미디어가 재생되지 않고 있습니다.

Example

[C++]

```
if (!m_ctlXnsDevice.isPlaying(m_hDevice))
{
```

```
m_ctlXnsDevice.Play(m_hDevice, 2, 1.0);
}
```

See Also

[Play](#)

HasAuthority

Description

애플리케이션이 미디어를 재생하기 위한 권한을 갖고 있는지 확인하는 함수입니다.

Syntax

```
BOOL HasAuthority(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- TRUE: 미디어 재생 권한이 있습니다.
- FALSE: 미디어 재생 권한이 없습니다.

Example

[C++]

```
long ret;
ret = m_ctlXnsDevice.GetControlCapability(m_hDevice, 1, XCTL_CAP_MEDIA_AUTHORITY);

DisplayErrCode("Check cap of MEDIA_AUTHORITY ", ret);
if (ret)
{
    ret = m_ctlXnsDevice.HasAuthority(m_hDevice);
    DisplayErrCode("HasAuthority", ret);
    if (ret)
    {
        ret = m_ctlXnsDevice.AcquireMediaAuthority(m_hDevice);
        DisplayErrCode("AcquireMediaAuthority", ret);
    }
}
```

```
    }
}
```

See Also

[AcquireMediaAuthority](#), [ReleaseMediaAuthority](#)

SetVideoFrameInfoReceived

Description

OpenMedia()를 호출하여 미디어 스트림을 받기 시작한 후, 이 함수를 호출하면, 수신되는 비디오 프레임의 프레임레이트, 비트레이트, 해상도, 인코딩 정보등을 이벤트로 받아올 수 있습니다. 수신된 프레임정보는 [OnVideoFrameInfoReceived](#) 이벤트로 전달됩니다.

Syntax

```
long SetVideoFrameInfoReceived
(
    long hDevice,
    long nControlID,
    long nMediaType,
    long bFlag
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] 비디오 control module의 control ID.

nMediaType

[in] 미디어 타입

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

bFlag

[in] 1일 경우, 비디오 정보 수신 활성화, 0인 경우 비디오 정보 수신 비활성화

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.

Example

[C++]

```
long ret;

ret = m_ctlXnsDevice.OpenMedia(m_hDevice, 2, XMEDIA_LIVE, 0, 0, &m_pMediaLiveSource);
ret = m_ctlXnsDevice.OpenMedia(m_hDevice, 2, XMEDIA_LIVE, TRUE);
```

See Also

[OpenMedia](#), [ControlID](#), [OnVideoFrameInfoReceived](#)

OpenText

Description

장비로부터 Text 스트림 수신을 시작합니다. 수신된 Text 스트림은 [OnNewMedia](#) 이벤트로 전달됩니다.

Syntax

```
long OpenText(
    long hDevice,
    long nControlID,
    long nMediaType,
    long tStart,
    long tEnd,
    long nTrackID
);
```

Parameters

hDevice

[in] Device handle. 0이 같은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Text device control module의 control ID.

nMediaType

[in] 미디어 타입

- XMEDIA_TEXT_LIVE (=5): Live
- XMEDIA_TEXT_PLAYBACK (=6): Playback
- XMEDIA_TEXT_BACKUP (=7): Backup replay (local PC에 Text 데이터를 저장함)

tStart

[in] 재생 시작 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

tEnd

[in] 재생 종료 시간. 4 바이트의 time_t 포맷임. (XMEDIA_BACKUP일때만 유효함)

nTrackID

[in] 중복구간 ID. XMEDIA_TEXT_LIVE 미디어 타입에서는 사용하지 않습니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.
- ERR_MEDIA_ALREADY_OPENED: 미디어가 이미 열려있음.

Example**[C++]**

```
long ret = m_ctlXnsDevice.OpenText(m_hDevice, 174, XMEDIA_TEXT_LIVE, 0, 0, 0);
```

See Also

[CloseText](#), [ControlID](#)

CloseText

Description

Text 스트림 전송을 중단합니다. Text 스트림은 Control ID와 미디어타입을 이용하여 구분합니다.

Syntax

```
long CloseText(
    long hDevice,
    long nControlID,
```

```
    long nMediaType  
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Text device control module의 control ID.

nMediaType

[in] 미디어 타입.

- XMEDIA_TEXT_LIVE (=5): Live
- XMEDIA_TEXT_PLAYBACK (=6): Playback
- XMEDIA_TEXT_BACKUP (=7): Backup replay (local PC에 스트림 데이터를 저장함)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

Example

[C++]

```
long ret = m_ctlXnsDevice.CloseText(m_hDevice, 174, XMEDIA_LIVE);
```

See Also

[OpenText](#), [ControlID](#)

PlayControl

참고

재생과 관련된 기능을 사용하기 위해서는 사전에 AcquireMediaAuthority()를 이용하여 권한을 얻어야 합니다. 또한, 사용 후에는 ReleaseMediaAuthority()를 이용하여 획득한 권한을 반환해야 합니다.

Play

Description

지정된 속도로 미디어를 재생합니다.

Syntax

```
long Play(
    long hDevice,
    long nControlID,
    double dSpeed
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

dSpeed

[in] 재생 속도.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
if (!m_ctlXnsDevice.isPlaying(m_hDevice))
    m_ctlXnsDevice.Play(m_hDevice, 2, 1.0);
```

See Also

[IsPlaying](#), [Pause](#), [ControlID](#)

Pause

Description

미디어 스트림 재생을 일시 중지합니다.

Syntax

```
long Pause(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
m_ctlXnsDevice.Pause(m_hDevice, 2);
```

See Also

[Play](#), [ControlID](#)

Seek

Description

미디어 스트림에서 특정 시간(위치)을 탐색합니다.

Syntax

```
long Seek(
    long hDevice,
    long nControlID,
    long tPos
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

tPos

[in] Target time. (4바이트의 time_t 포맷)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long tPos = m_ctlXnsDevice.DateToTimet(2010, 9, 6, 15, 0, 0);
m_ctlXnsDevice.Seek(m_hDevice, 2, tPos);
```

See Also

[DateToTimet](#), [ControlID](#)

FrameAdvance

Description

다음(next) I-frame을 가져옵니다.

Syntax

```
long FrameAdvance(
    long hDevice,
    long nControlID,
    long nDelta
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nDelta

[in] Skip할 프레임 수. 이 값이 0보다 작으면, 이전 프레임을 가져옴.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
m_ctlXnsDevice.FrameAdvance(m_hDevice, 2, 1); // forward
m_ctlXnsDevice.FrameAdvance(m_hDevice, 2, -1); // backward
```

See Also

[Play](#), [ControlID](#)

SearchCalendar

Description

특정 달에서 영상이 녹화된 날짜들을 가져옵니다. 이 함수의 결과는 OnSearchCalendar 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long SearchCalendar(  
    long hDevice,  
    long nControlID,  
    long tStart  
)
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

tStart

[in] 특정 달의 첫째 날. (4 바이트 time_t 포맷)

예를 들어, 애플리케이션이 2010년 1월의 녹화 정보를 얻으려고 한다면, tStart는 "2010년 1월 1일 00:00:00"을 DateToTimet()를 이용하여 time_t 포맷으로 변형한 값이 됩니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```
[C++]  
long tStart = m_ctrlXnsDevice.DateToTimet(2010, 9, 1, 0, 0, 0);           // 2010년 9월 1일  
long nUTCtime = m_ctrlXnsDevice.LocalTimeToDeviceUTC(m_hDevice, tStart); // UTC 시간 적용  
long ret = m_ctrlXnsSdkDevice.AcquireMediaAuthority(m_hDevice);  
ret = m_ctrlXnsDevice.SearchCalendar(m_hDevice, 1, nUTCtime);  
DisplayErrCode("SearchCalendar", ret);
```

See Also

[OnSearchCalendar](#), [DateToTimet](#), [SearchDay](#), [ControlID](#), [LocalTimeToDeviceUTC](#),
[AcquireMediaAuthority](#)

SearchOverLap

Description

하루 중 녹화된 데이터의 중복구간 정보를 가져옵니다. 이 함수의 결과는 OnSearchOverLap 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long SearchOverLap(
    long hDevice,
    long tStart,
    long tEnd
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

tStart

[in] 검색 시작 시간. (UTC time_t 포맷)

tEnd

[in] 검색 종료 시간. (UTC time_t 포맷)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```
[C++]
long tStart = m_ctrlXnsDevice.DateToTimet(2012, 9, 04, 0, 0, 0);
long tEnd = m_ctrlXnsDevice.DateToTimet(2012, 9, 04, 23, 59, 59);

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tEnd);

long ret = m_ctrlXnsDevice.SearchOverLap(m_hDevice, tUTCStart, tUTCEnd);

DisplayErrCode("SearchOverLap", ret);
```

See Also

[OnSearchCalendar](#), [OnSearchOverLap](#), [DateToTimet](#), [SearchCalendar](#), [SearchDay](#),
[LocalTimeToDeviceUTC](#), [AcquireMediaAuthority](#), APPENDIX C. [UTC Time](#)

GetStartRecordingTime

Description

DVR의 녹화된 데이터의 시작시간을 가져옵니다. 이 함수의 결과는 OnGetStartRecordingTime 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long GetStartRecordingTime(
    long hDevice,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.

Example

```
[C++]
long ret = m_ctlXnsDevice.GetStartRecordingTime(m_hDevice);

DisplayErrCode("GetStartRecordingTime ", ret);
```

See Also

[OnGetStartRecordingTime](#), [GetEndRecordingTime](#), [OnGetEndRecordingTime](#)

GetEndRecordingTime

Description

DVR의 녹화된 데이터의 종료시간을 가져옵니다. 이 함수의 결과는 OnGetEndRecordingTime 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long GetEndRecordingTime(
    long hDevice,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.

Example

```
[C++]
long ret = m_ctlXnsDevice.GetEndRecordingTime(m_hDevice);

DisplayErrCode("GetEndRecordingTime ", ret);
```

See Also

[OnGetEndRecordingTime](#), [GetStartRecordingTime](#), [OnGetStartRecordingTime](#)

SearchDay

Description

하루 중 녹화된 데이터 정보를 가져옵니다. 이 함수의 결과는 OnSearchDay 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long SearchDay(
    long hDevice,
```

```

        long nControlID,
        long tStart,
        long tEnd,
        long nRecType
    );

```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

tStart

[in] 재생 시작 시간. (UTC time_t 포맷)

tEnd

[in] 재생 종료 시간. (UTC time_t 포맷)

nRecType

[in] Recording type.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```

[C++]
long tStart = m_ctrlXnsDevice.DateToTimet(2010, 9, 06, 0, 0, 0);
long tEnd = m_ctrlXnsDevice.DateToTimet(2010, 9, 06, 23, 59, 59);

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tEnd);

long ret = m_ctrlXnsDevice.SearchDay(m_hDevice, 2, tUTCStart, tUTCEnd, REC_TYPE_ALL);

DisplayErrCode("SearchDay", ret);

```

See Also

[OnSearchDay](#), [DateToTimet](#), [SearchCalendar](#), [LocalTimeToDeviceUTC](#), [ControlID](#), APPENDIX C.
[UTC Time](#)

GetMotionGrid

Description

DVR의 Motion Grid 정보를 가져옵니다. 이 함수의 결과는 OnGetMotionGrid 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long GetMotionGrid(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```
[C++]
long ret = m_ctlXnsDevice.GetMotionGrid(m_hDevice, m_nControlID);

DisplayErrCode("GetMotionGrid", ret);
```

See Also

[OnGetMotionGrid](#), [MotionSearch](#), [OnMotionSearch](#),

MotionSearch

Description

설정한 Grid에서 녹화된 데이터 정보를 가져옵니다. 이벤트 타입은 Motion 이벤트입니다. 이 함수의 결과는 OnMotionSearch 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long MotionSearch(
    long hDevice,
    long nControlID,
    long tStart,
    long tEnd,
    long nGridWidth,
    long nGridHeight,
    long pGridArray
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

tStart

[in] 검색 시작 시간. (UTC time_t 포맷)

tEnd

[in] 검색 종료 시간. (UTC time_t 포맷)

nGridWidth

[in] 가로 Grid 개수(최대 64).

nGridHeight

[in] 세로 Grid 개수(최대 64).

pGridArray

[in] nGridWidth X nGridHeight 개수만큼의 char 배열값(최대 64 X 64). 검색을 원하는 Grid에 '1'로 마킹.

Ex) 3 X 3 의 pGridArray일경우 화면영역의 Grid는 아래와 같음

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]
[2][0]	[2][1]	[2][2]

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```
[C++]
long tStart = m_ctrlXnsDevice.DateToTimet(2010, 9, 06, 0, 0, 0);
long tEnd = m_ctrlXnsDevice.DateToTimet(2010, 9, 06, 23, 59, 59);
char m_pGridArray[64][64];

for(int i = 0; i < m_nGridHeight; i++)
{
    for(int j = 0; j < m_nGridWidth; j++)
    {
        m_pGridArray[i][j] = '1';
    }
}

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tEnd);

long ret = m_ctrlXnsDevice.MotionSearch(m_hDevice, 2, tUTCStart, tUTCEnd, m_nGridWidth,
m_nGridHeight, (long)m_pGridArray);

DisplayErrCode("MotionSearch", ret);
```

See Also

[OnMotionSearch](#), [GetMotionGrid](#), [OnGetMotionGrid](#), [DateToTimet](#), [SearchCalendar](#),

[LocalTimeToDeviceUTC](#), [ControlID](#), APPENDIX C. [UTC Time](#)

SearchTextCalendar

Description

특정 달에서 Text가 저장된 날짜들을 가져옵니다. 이 함수의 결과는 OnSearchTextCalendar 이벤트를 통해서 받을 수 있습니다.

Syntax

```
long SearchTextCalendar(
    long hDevice,
    long nControlID,
    long tStart
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Text device control module의 control ID.

tStart

[in] 특정 달의 첫째 날. (4 바이트 time_t 포맷)

예를 들어, 애플리케이션이 2016년 1월의 녹화 정보를 얻으려고 한다면, tStart는 "2016년 1월 1일 00:00:00"을 DateToTimet()를 이용하여 time_t 포맷으로 변형한 값이 됩니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long tStart = m_ctlXnsDevice.DateToTimet(2016, 1, 1, 0, 0, 0);           // 2016년 1월 1일
long nUTCTime = m_ctlXnsDevice.LocalTimeToDeviceUTC(m_hDevice, tStart); // UTC 시간 적용
```

```

long nControlCount = m_ctlXnsDevice.GetControlCount(m_hDevice, XCTL_ALL);
long nControlID = 0;
for (int i=1; i<nControlCount; i++)
{
    if( m_ctlXnsDevice.GetControlType(m_hDevice, i) == XCTL_TEXT )
    {
        nControlID = i;
        break;
    }
}

long ret = m_ctlXnsDevice.SearchTextCalendar(m_hDevice, i, nUTCtime);
DisplayErrCode("SearchTextCalendar", ret);

```

See Also

[OnSearchTextCalendar](#), [DateToTimet](#), [SearchText](#), [ControlID](#), [LocalTimeToDeviceUTC](#),

SearchText

Description

저장된 Text 정보를 가져옵니다. 이 함수의 결과는 OnSearchTextList, OnEndSearchText 이벤트를 통해서 받을 수 있습니다.

Syntax

```

long SearchText(
    long hDevice,
    long nControlID,
    long tStart,
    long tEnd,
    long nTrackID,
    bool bIsAllMatch,
    bool bIsCaseSensitive,
    BSTR szText
);

```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Text device control module의 control ID.

tStart

[in] 시작 시간. (UTC time_t 포맷)

tEnd

[in] 종료 시간. (UTC time_t 포맷)

nTrackID

[in] 중복구간 ID

bIsAllMatch

[in] 전체단어 일치 여부를 결정하는 플래그..

bIsCaseSensitive

[in] 대소문자 일치 여부를 결정하는 플래그.

szText

[in] 검색할 text.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```
[C++]
long tStart = m_ctrlXnsDevice.DateToTimet(2016, 6, 1, 0, 0, 0);
long tEnd = m_ctrlXnsDevice.DateToTimet(2010, 6, 1, 23, 59, 59);

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tEnd);

long nControlID = 0;
for (int i=1; i<nControlCount; i++)
{
    if( m_ctrlXnsDevice.GetControlType(m_hDevice, i) == XCTL_TEXT )
    {
        nControlID = i;
```

```

        break;
    }
}

long ret = m_ctlXnsDevice.SearchText(m_hDevice, nControlID, tUTCStart, tUTCEnd, nTrackID,
false, false, _T("coke"));

DisplayErrCode("SearchText", ret);

```

See Also

[OnSearchTextList](#), [OnEndSearchText](#), [DateToTimet](#), [SearchTextCalendar](#), [LocalTimeToDeviceUTC](#),
[ControlID](#), APPENDIX C. [UTC Time](#), ,

CancelSearchText

Description

Text 검색을 취소합니다.

Syntax

```
long CancelSearchText(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long errCode = m_ctlXnsDevice.CancelSearchText(m_hDevice);
```

See Also

[SearchText](#)

Control Audio and Alarm-out

SetTalk

Description

Talk 기능을 켜거나 끄는 함수입니다. Talk는 장비에서 오디오를 재생하는 기능입니다. 애플리케이션은 SendAudioData() / SendAudioData2()를 이용하여 재생하고자 하는 오디오를 전송해야 합니다. 이 기능은 라이브 스트리밍(live streaming) 모드에서만 가능하며, control module이 XCTL_CAP_TALK capability를 지원해야 합니다.

Syntax

```
long SetTalk(
    long hDevice,
    long nControlID,
    BOOL bState
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

bState

[in] Talk 기능의 상태.

- 1: Talk 기능을 시작함(on).
- 0: Talk 기능을 종료함(off).

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.SetTalk(m_hDevice, 2, 1);
```

See Also

[SetListen](#), [SendAudioData](#), [SendAudioData2](#), [ControlID](#)

SetListen

Description

Listen기능을 On또는 Off시킬 경우 XnsSDKWindow컨트롤의 Listen 아이콘을 활성화/비활성화 시키는 함수입니다. Listen은 애플리케이션에서 오디오를 재생하는 기능입니다. 이 기능을 사용 하려면, control module에서 XCTL_CAP_LISTEN capability를 지원해야 합니다.

Syntax

```
long SetListen(
    long hDevice,
    long nControlID,
    long nMediaType,
    BOOL bState
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nMediaType

[in] 미디어 타입

- XMEDIA_LIVE (=1) : Live
- XMEDIA_PLAYBACK (=2) : Playback

bState

[in] Listen 기능의 상태.

- 1: Listen 아이콘 활성화(enable)
- 0: Listen 아이콘 비활성화(disable)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.SetListen(m_hDevice, 2, XMEDIA_LIVE, 1);
```

See Also

[SetSound](#), [SetTalk](#), [ControlID](#)

SendAudioData

Description

장비에 오디오 데이터를 전송합니다. 이 함수는 Talk 기능을 사용할 때, 애플리케이션이 오디오 데이터를 장비에 전송하기 위해 사용합니다. 이때, 오디오 데이터는 각 장비에 맞게 인코딩(encoding)되어 있어야 합니다. 이 함수는 라이브 모드(live mode)에서만 유효합니다.

Syntax

```
long SendAudioData(
    long hDevice,
    long hData,
    long nDataSize,
    long nTimeStamp
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

hData

[in] 오디오 데이터가 저장된 메모리를 가리키는 포인터.

nDataSize

[in] 전송할 오디오 데이터의 길이.

nTimeStamp

[in] 오디오 재생 시간. (단위: millisecond)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.SendAudioData(m_hDevice, pAudio, nAudioSize, nTimeStamp);
```

See Also

[SetTalk](#)

SendAudioData2

Description

장비에 오디오 데이터를 전송합니다. 이 함수는 Talk 기능을 사용할 때, 애플리케이션이 오디오 데이터를 장비에 전송하기 위해 사용합니다. 이 함수는 라이브 모드(live mode)에서만 유효합니다.

Syntax

```
long SendAudioData2(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.SendAudioData2(m_hDevice, nControlID);
```

See Also

[SetTalk](#), [StopAudioData2](#)

StopAudioData2

Description

SendAudioData2 함수를 중지합니다.

Syntax

```
long StopAudioData2( long hDevice );
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long ret = m_ctlXnsDevice.StopAudioData2(m_hDevice);
```

See Also

[SetTalk](#), [SendAudioData2](#)

SetAlarm

Description

Alarm-out 장비(digital-out device라고도 함)를 켜고 끄는 함수입니다. 이 함수를 사용하려면 control module이 XCTL_CAP_ALARM_ONOFF capability를 지원해야 합니다.

Syntax

```
long SetAlarm(
    long hDevice,
    long nControlID,
    BOOL bState
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

bState

[in] Alarm 장비의 상태.

- 1: Alarm-out device on
- 0: Alarm-out device off

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.SetAlarm(m_hDevice, 30, 1);
```

See Also

[ControlID](#)

Control PTZ

ControlPtz

Description

카메라를 사전에 정의된 PTZ 위치로 이동시킵니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long ControlPtz(
    long hDevice,
    long nControlID,
    long nPtzCommand,
    long nSpeed
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nPtzCommand

[in] PTZ 명령. ([표 2.8](#) 참조)

nSpeed

[in] *nPtzCommand*가 속도와 관련된 명령인 경우, 이 값은 PTZ 구동 속도를 나타냄. (1~100) 이 값은 카메라가 XCTL_CAP_PTZ_SPEED capability를 지원하는 경우에만 유효함.

표 2.8 PTZ Command

Command	Value	Description
XPTZ_UP	1	Tilt up
XPTZ_DOWN	2	Tilt down
XPTZ_LEFT	3	Pan left
XPTZ_RIGHT	4	Pan right

XPTZ_UPLEFT	5	Tilt up and pan left
XPTZ_UPRIGHT	6	Tilt up and pan right
XPTZ_DOWNLEFT	7	Tilt down and pan left
XPTZ_DOWNRIGHT	8	Tilt down and pan right
XPTZ_ZOOMIN	9	Zoom in
XPTZ_ZOOMOUT	10	Zoom out
XPTZ_STOP	11	Stop the PTZ moving
XPTZ_FOCUS_NEAR	12	Focus near
XPTZ_FOCUS_FAR	13	Focus far
XPTZ_FOCUS_STOP	14	Stop focus moving
XPTZ_IRIS_OPEN	15	Open iris
XPTZ_IRIS_CLOSE	16	Close iris

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long bIsCap = m_ctlXnsDevice.GetControlCapability(m_hDevice, 2, XCTL_CAP_PTZ_SPEED);
if (bIsCap)
    m_ctlXnsDevice.ControlPtz(m_hDevice, 2, XPTZ_UPLEFT, nSpeed);
else
    m_ctlXnsDevice.ControlPtz(m_hDevice, 2, XPTZ_UPLEFT, 10);
```

See Also

[ControlPtzFreeMove](#), [ControlID](#)

ControlPtzFreeMove

Description

카메라를 임의의 PTZ 위치로 이동시킵니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있을 때만 유효합니다.

Syntax

```
long ControlPtzFreeMove(
    long hDevice,
    long nControlID,
    long nX,
    long nY
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nX

[in] X축 방향으로의 속도. 1~100 사이의 값을 가지며, 값이 클수록 빠른 속도임.

nY

[in] Y축 방향으로의 속도. 1~100 사이의 값을 가지며, 값이 클수록 빠른 속도임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.ControlPtzFreeMove(m_hDevice, 2, 10, 22);
```

See Also

[ControlPtz](#), [ControlID](#)

ControlPtzSpeedZoomIn

Description

지정된 속도로 zoom-in 작업을 수행합니다. 이 함수는 애플리케이션이 카메라로부터 스트림 데이터를 수신하고 있는 동안에만 유효합니다.

Syntax

```
long ControlPtzSpeedZoomIn(
    long hDevice,
    long nControlID,
    long nSpeed
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nSpeed

[in] Zoom 이동 속도. 1~100 사이의 값을 가지며 숫자가 클수록 빠른 스피드를 나타냄.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.ControlPtzSpeedZoomIn(m_hDevice, 2, 6);
```

See Also

[ControlPtzSpeedZoomOut](#), [ControlID](#)

ControlPtzSpeedZoomOut

Description

지정된 속도로 zoom-out 작업을 수행합니다. 이 함수는 애플리케이션이 카메라로부터 스트림 데이터를 수신하고 있는 동안에만 유효합니다.

Syntax

```
long ControlPtzSpeedZoomOut(
    long hDevice,
    long nControlID,
    long nSpeed
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nSpeed

[in] Zoom 이동 속도. 1~4 사이의 값을 가지며 숫자가 클수록 빠른 스피드를 나타냄.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long ret = m_ctlXnsDevice.ControlPtzSpeedZoomOut(m_hDevice, 2, 3);
```

See Also

[ControlPtzSpeedZoomIn](#), [ControlID](#)

ControlPPtz

Description

Control Power PTZ. 현재 위치에서 pan, tilt, zoom을 이동하는 함수입니다. 이 함수는 삼성 카메라 중 몇 개 모델에서만 사용할 수 있습니다. 모든 파라미터는 standard VGA image로 크기가 조정되어야 합니다. 이미지에서 왼쪽 상단은 (0,0), 오른쪽 하단은 (640, 480)의 좌표를 갖습니다. 만약, 애플리케이션이 nX 와 nY 를 320, 250으로 설정했다면, 카메라는 아래쪽 방향(tilt down)으로 약간 움직입니다. nZ 은 줌(zoom factor)입니다. 카메라는 이 값이 양수이면 zoom-in, 음수이면 zoom-out 됩니다.

이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long ControlPPtz(
    long hDevice,
    long nControlID,
    long nX,
    long nY,
    long nZ
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nX

[in] X축 이동. (0~640 사이의 값을 가짐)

nY

[in] Y축 이동. (0~480 사이의 값을 가짐)

nZ

[in] Zoom factor.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
BOOL bIsCap = m_ctlXnsDevice.GetControlCapability(m_hDevice, 2, XCTL_CAP_PTZ_POWER);
DisplayLong("GetPtz XCTL_CAP_PTZ_POWER Cap", bIsCap);

long nErrCode = m_ctlXnsDevice.ControlPPtz(m_hDevice, 2, 50, 50, 2);
DisplayErrCode("OnPowerPtz", nErrCode);
```

See Also

[ControlPtzFreeMove](#), [ControlID](#)

ControlMenu

Description

카메라의 OSD 메뉴를 제어합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long ControlMenu(
    long hDevice,
    long nControlID,
    long nMenuCommand
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nMenuCommand

[in] OSD의 메뉴를 제어하는 명령. ([표 2.9](#) 참조)

표 2.9 Menu Control Command

Command	Value	Description
XMENU_UP	1	Menu up
XMENU_DOWN	2	Menu down
XMENU_LEFT	3	Menu left or prev menu
XMENU_RIGHT	4	Menu right or next menu
XMENU_ENTER	5	Menu enter, select
XMENU_ON	6	Menu on
XMENU_OFF	7	Menu off
XMENU_CANCEL	8	Cancel. (Changes are not saved.)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Remark

OSD 메뉴 제어 기능은 일부 지원 기기(예. SNP-1000A)에서만 제공합니다.

Example

[C++]

```
BOOL bIsCap = m_ctlXnsDevice.GetControlCapability(m_hDevice, 2, XCTL_CAP_CAM_MENU);
DisplayLong("GetPtz XCTL_CAP_CAM_MENU Cap", bIsCap);

long nErrCode = m_ctlXnsDevice.ControlMenu(m_hDevice, 2, XMENU_ENTER);
DisplayErrCode("OnPtzMenuEnter", nErrCode);
```

See Also

[ControlID](#)

ExecuteHPtz

Description

프리셋(preset), 오토팬(autopan), 스캔(scan), 패턴(pattern) 기능을 수행합니다. 이 함수는 애플리케이션에서 사용하는 주요 기능입니다.

리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long ExecuteHPtz(
    long hDevice,
    long nControlID,
    long nType,
    long nNumber
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nType

[in] 카메라 기능.

- XHPTZ_PRESET(1): Preset
- XHPTZ_AUTOPAN(2): Autopan(Swing)
- XHPTZ_SCAN(3): Scan
- XHPTZ_PATTERN(4): Pattern
- XHPTZ_SWING(5): Swing
- XHPTZ_TOUR(6): Tour

nNumber

[in] 카메라가 갖고 있는 기능(프리셋, 오토팬, 스캔, 패턴) 리스트의 인덱스. (1에서 시작함) 예를 들어, 프리셋이 여러 개 지정되어 있는 경우, 프리셋 리스트가 생성되면 사용자는 그 리스트의 인덱스를 지정해야 함.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
BOOL bIsCap = m_ctlXnsDevice.GetControlCapability(m_hDevice, 2, XCTL_CAP_PTZ_SCAN);
DisplayLong("GetPtz XCTL_CAP_PTZ_SCAN Cap", bIsCap);

long nErrCode = m_ctlXnsDevice.ExecuteHPtz(m_hDevice, 2, XHPTZ_SCAN, 1);
DisplayErrCode("OnScan", nErrCode);
```

See Also[ControlID](#)

GetPresetList

Description

프리셋 리스트를 가져옵니다. 프리셋은 사용자가 정의한 카메라 위치를 말합니다. 설정할 수 있는 프리셋의 수는 카메라마다 다르며, XNS ActiveX 라이브러리는 20개까지 지원합니다. 이 함수의 수행 결과는 OnGetPresetList 이벤트를 통해서 전달됩니다.

이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long GetPresetList(
    long hDevice,
    long nControlID
);
```

Parameters*hDevice*

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long m_hPreset = m_ctlXnsDevice.GetPresetList(m_hDevice, 2);
```

See Also

[OnGetPresetList](#), [ControlID](#)

SetPreset

Description

현재 카메라의 위치를 프리셋으로 설정합니다. 이 위치는 카메라에 저장됩니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long SetPreset(
    long hDevice,
    long nControlID,
    long nNumber,
    LPCTSTR szName
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nNumber

[in] 저장할 프리셋 번호. (1부터 시작함)

szName

[in] 프리셋 이름. 카메라는 프리셋 위치를 이 이름과 함께 저장함. (English only)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.

- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.SetPreset(m_hDevice, 2, 10, "Preset 10");
```

See Also

[DeletePreset](#), [ControlID](#)

DeletePreset

Description

저장된 프리셋을 삭제합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long DeletePreset(
    long hDevice,
    long nControlID,
    long nNumber
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nNumber

[in] 삭제할 프리셋 번호. (0부터 시작함.)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.DeletePreset(m_hDevice, 2, 10);
```

See Also

[SetPreset](#), [DeletePresetAll](#), [ControlID](#)

DeletePresetAll

Description

프리셋을 모두 삭제합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long DeletePresetAll(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.DeletePresetAll(m_hDevice, 2);
```

See Also

[DeletePreset](#), [SetPreset](#), [ControlID](#)

GetPresetCount

Description

설정된 프리셋의 개수를 가져옵니다. 이 값은 프리셋 이름과 정보를 얻기 위해 사용합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long GetPresetCount(long hPresetList);
```

Parameters

hPresetList

[in] 프리셋 리스트의 핸들. 이 핸들은 OnGetPresetList 이벤트를 통해서 얻을 수 있음.

Return Value

카메라에 설정된 프리셋의 총 개수.

Remarks

이 함수는 OnGetPresetList 이벤트의 이벤트 핸들러 내에서만 유효합니다.

Example

[C++]

```
// 이 함수는 OnGetPresetList 이벤트가 발생하면 호출되는 이벤트 핸들러 함수입니다.
void CTestSampleDlg::OnGetPresetListXnssdkdevicectrl1(long nDeviceID, long nControlID, long hPresetListList)
{
    char str[100];
    long nNumber;
    long nPresetCount;
    CString strPresetName;

    nPresetCount = m_ctlXnsDevice.GetPresetCount(hPresetListList);
    sprintf(str, "Event OnGetPresetList.. nDevice[%d], nControlId[%d], resetCount[%d]\n",
            nDeviceID, nControlID, nPresetCount);
    OutputDebugString(str);
```

```

for(int i=0; i<nPresetCount; i++)
{
    strPresetName = m_ctlXnsDevice.GetPreset(hPresetListList, i, &nNumber);
    sprintf(str, "PresetName[%s], nPresetNumber[%d]\n", strPresetName,
    nNumber);
    OutputDebugString(str);
}
}

```

See Also

[GetPresetList](#), [OnGetPresetList](#), [GetPreset](#)

GetPreset

Description

XnsSdkDevice에 저장된 프리셋 리스트에서 주어진 인덱스에 대응하는 프리셋의 번호와 이름을 반환합니다.

Syntax

```

CString GetPreset(
    long hPresetList,
    long nPresetIndex,
    long* pnNumber
);

```

Parameters

hPresetList

[in] 프리셋 리스트의 핸들. 이 핸들은 OnGetPresetList 이벤트를 통해서 얻을 수 있음.

nPresetIndex

[in] 프리셋 정보를 얻기 위한 프리셋 인덱스. (1부터 시작함.)

pnNumber

[out] XnsSdkDevice에 저장된 프리셋 인덱스.

Return Value

카메라에 저장된 프리셋 이름.

Remarks

이 함수는 OnGetPresetList 이벤트의 이벤트 핸들러 내에서만 유효합니다.

Example

[C++]

```
// 이 함수는 OnGetPresetList 이벤트가 발생하면 호출되는 이벤트 핸들러 함수입니다.

void CTestSampleDlg::OnGetPresetListXnssdkdevicectrl1(long nDeviceID, long nControlID, long
hPresetListList)
{
    char str[100];
    long nNumber;
    long nPresetCount;
    CString strPresetName;

    nPresetCount = m_ctlXnsDevice.GetPresetCount(hPresetListList);
    sprintf(str, "Event OnGetPresetList.. nDevice[%d], nControlId[%d], resetCount[%d]\n",
            nDeviceID, nControlID, nPresetCount);
    OutputDebugString(str);

    for(int i=0; i<nPresetCount; i++)
    {
        strPresetName = m_ctlXnsDevice.GetPreset(hPresetListList, i, &nNumber);
        sprintf(str, "PresetName[%s], nPresetNumber[%d]\n", strPresetName,
                nNumber);
        OutputDebugString(str);
    }
}
```

See Also

[GetPresetList](#), [OnGetPresetList](#)

GetPtzPos

Description

카메라의 현재 절대 좌표 위치를 요청합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다. 절대좌표 위치는 OnGetPtzPos 이벤트 핸들러를 통해 읽어올 수 있습니다.

Syntax

```
long GetPtzPos(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
m_ctlXnsDevice.GetPtzPos(m_hDevice, 2);
```

See Also

[OnGetPtzPos](#), [SetPtzPos](#), [ControlID](#)

SetPtzPos

Description

카메라의 현재 절대좌표 위치를 변경합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long SetPtzPos(
    long hDevice,
```

```

        long nControlID,
        long nPan,
        long nTilt,
        long nZoom
    );

```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

nPan

[in] Pan 절대좌표값

nTilt

[in] Tilt 절대좌표값

nZoom

[in] Zoom 절대좌표값

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```

long nRet = m_ctlXnsDevice.SetPtzPos(m_hDevice, 2, m_nPan, m_nTilt, m_nZoom);
if(nRet != ERR_SUCCESS)
{
    TRACE (_T("SetPtzPos() fail:: nRet = %d(%s)\n"),
    nRet, m_ctlXnsDevice.GetErrorString(nRet));
}

```

See Also

[OnGetPtzPos](#), [GetPtzPos](#), [ControlID](#)

GetPtzPosNormalize

Description

카메라의 현재 좌표 위치를 요청합니다. 이 함수는 SUNAPI 프로토콜로 접속 시 사용 가능하며, 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다. 좌표 위치는 OnGetPtzPosNormalize 이벤트 핸들러를 통해 읽어올 수 있습니다.

Syntax

```
long GetPtzPosNormalize(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
m_ctlXnsDevice.GetPtzPosNormalize(m_hDevice, 2);
```

See Also

[OnGetPtzPosNormalize](#), [SetPtzPosNormalize](#), [ControlID](#)

SetPtzPosNormalize

Description

카메라의 현재 좌표 위치를 변경합니다. 이 함수는 SUNAPI 프로토콜로 접속 시 사용 가능하며, 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long SetPtzPosNormalize(  
    long hDevice,  
    long nControlID,  
    float fPan,  
    float fTilt,  
    float fZoom  
)
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

fPan

[in] Pan 좌표값

fTilt

[in] Tilt 좌표값

fZoom

[in] Zoom 좌표값

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long nRet = m_ctlXnsDevice.SetPtzPosNormalize(m_hDevice, 2, m_fPan, m_fTilt, m_fZoom);
if(nRet != ERR_SUCCESS)
{
    TRACE (_T("SetPtzPosNormalize() fail:: nRet = %d(%s)\n"),
        nRet, m_ctlXnsDevice.GetErrorString(nRet));
}
```

See Also

[OnGetPtzPosNormalize](#), [GetPtzPosNormalize](#), [ControlID](#)

AreaZoom

Description

카메라의 AreaZoom 기능을 설정합니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long AreaZoom(
    long hDevice,
    long nControlID,
    long nStartX,
    long nStartY,
    long nEndX,
    long nEndY,
    long nWindowWidth,
    long nWindowHeight,
    long nImageWidth,
    long nImageHeight
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임

nControlID

[in] Control ID

nStartX

[in] 시작 좌표 X값

nStartY

[in] 시작 좌표 Y값

nEndX

[in] 끝 좌표 X값

nEndY

[in] 끝 좌표 Y값

nWindowWidth

[in] 윈도우 컨트롤의 폭

nWindowHeight

[in] 윈도우 컨트롤의 높이

nImageWidth

[in] 재생 영상의 해상도 폭

nImageHeight

[in] 재생 영상의 해상도 높이

Return Value

- ERR_SUCCESS: 함수 수행에 성공함
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터
- ERR_NOT_CONNECT: 장비가 연결되지 않음
- ERR_NO_PERMISSION: 권한이 없음

Example

[C++]

```
void CMy01_LiveDlg::OnLButtonDownXnssdkwindowctrl(long nFlags, long nX, long nY)
```

```
{
```

```
// TODO: Add your message handler code here
```

```
WIN_LOG(_T("On LButtonUp XNSWindow1"));
```

```
m_nStartX = nX;
```

```
m_nStartY = nY;
```

```
}
```

```
void CMy01_LiveDlg::OnLButtonUpXnssdkwindowctrl(long nFlags, long nX, long nY)
```

```
{
```

```

long nImageWidth = m_ctlXnsWindow[0].GetImageWidth();
long nImageHeight = m_ctlXnsWindow[0].GetImageHeight();

CRect rt;
m_ctlXnsWindow[0].GetClientRect(rt);
long nWindowWidth = rt.Width();
long nWindowHeight = rt.Height();

m_nEndX = nX;
m_nEndY = nY;

WIN_LOG(_T("On LButtonUp XNSWindow1 flag=%d, x=%d, y=%d, imagew=%d,
imgaeh=%d, windoww=%d, windowh=%d"),
nFlags, nX, nY, nImageWidth, nImageHeight, nWindowWidth, nWindowHeight);

long nRet = m_ctlXnsDevice.AreaZoom(m_hDevice, 2, m_nStartX, m_nStartY,
m_nEndX, m_nEndY,
nWindowWidth, nWindowHeight,
nImageWidth, nImageHeight);

if(nRet != ERR_SUCCESS)
{
    WIN_LOG(_T("AreaZoom() fail:: nRet = %d(%s)\n"), nRet,
m_ctlXnsDevice.GetErrorString(nRet));
}
}
}

```

See Also

[Zoom1X](#), [ControlID](#)

Zoom1X

Description

Zoom1X는 카메라의 AreaZoom 기능을 설정하는 함수로, 현재 카메라의 좌표에서 Zoom 설정 만을 1배로 돌립니다. 이 함수는 애플리케이션이 카메라로부터 미디어 스트림을 수신하고 있는 동안에만 유효합니다.

Syntax

```
long Zoom1X(
    long hDevice,
    long nControlID,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_PTZ_CONFLICT: PTZ 작동 시, 충돌 발생.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

```
[C++]
long nRet = m_ctlXnsDevice.Zoom1X(m_hDevice, 2);

if(nRet != ERR_SUCCESS)
{
    WIN_LOG(_T("AreaZoom() fail:: nRet = %d(%s)\n"),
            nRet, m_ctlXnsDevice.GetErrorString(nRet));
}
```

See Also

[AreaZoom](#), [ControlID](#)

GetCameraCount

Description

타임라인(timeline)에 있는 카메라의 수를 반환합니다.

Syntax

```
long GetCameraCount(long hTimeline);
```

Parameters

hTimeline

[in] OnSearchDay 이벤트를 통해서 얻은 타임라인 핸들.

Return Value

시스템에 연결된 카메라 중 지정한 날짜에 녹화된 데이터가 있는 카메라의 개수.

Remarks

이 함수는 OnSearchDay 이벤트의 이벤트 핸들러 내에서만 유효합니다.

Example

[C++]

```
// 이 함수는 OnSearchDay 이벤트가 발생했을 때 호출되는 이벤트 핸들러입니다.
void CTestSampleDlg::OnSearchDayXnssdkdevicectrl1(long nDeviceID, long nControlID, long
hTimeline)
{
    long CamCount = m_ctlXnsDevice.GetCameraCount(hTimeline);
    // ...
}
```

See Also

[OnSearchDay](#), [GetTimelineCount](#)

GetTimelineCount

Description

타임라인에 있는 녹화된 데이터 블록(block)의 수. 이 값은 GetStartTime(), GetEndTime(), GetRecordingType()의 파라미터로 사용됩니다.

Syntax

```
long GetTimelineCount(
    long hTimeline,
    long nCameraIndex
);
```

Parameters

hTimeline

[in] OnSearchDay 이벤트를 통해서 얻은 타임라인 핸들.

nCameraIndex

[n] nCameraIndex GetCameraCount()를 통해서 가져온 카메라 리스트의 index.

Return Value

타임라인 내에서 녹화된 데이터 블록의 수.

Remarks

이 함수는 OnSearchDay 이벤트의 이벤트 핸들러 내에서만 유효합니다.

Example

[C++]

```
// 이 함수는 OnSearchDay 이벤트가 발생했을 때 호출되는 이벤트 핸들러입니다.
void CTestSampleDlg::OnSearchDayXnssdkdevicectrl1(long nDeviceID, long nControlID, long
hTimeline)
{
    long CamCount = m_ctlXnsDevice.GetCameraCount(hTimeline);
    long TimelineCount = m_ctlXnsDevice.GetTimelineCount(hTimeline, 0);

    CString strRecInfo;
    long tStart, tEnd;
    long nRecType;
    long nRecId;
    long StartYear, StartMon, StartDay, EndYear, EndMon, EndDay;
    long StartHour, StartMin, StartSec, EndHour, EndMin, EndSec;

    OutputDebugString("<On UTC time>");
    for(int i=0; i<TimelineCount; i++)
    {
        m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
```

```

&nRecType, &nRecId);

        m_ctlXnsDevice.TimetToDate(tStart, &StartYear, &StartMon, &StartDay,
&StartHour, &StartMin, &StartSec);

        m_ctlXnsDevice.TimetToDate(tEnd, &EndYear, &EndMon, &EndDay, &EndHour,
&EndMin, &EndSec);

        // ...

    }

    OutputDebugString("<On Local time>");

    for(i=0; i<TimelineCount; i++)
    {

        m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
&nRecType, &nRecId);

        m_ctlXnsDevice.TimetToDate(tStart, &StartYear, &StartMon, &StartDay,
&StartHour, &StartMin, &StartSec);

        m_ctlXnsDevice.TimetToDate(tEnd, &EndYear, &EndMon, &EndDay, &EndHour,
&EndMin, &EndSec);

        // ...

    }

}

```

See Also

[OnSearchDay](#), [GetCameraCount](#), [GetTimeline](#)

GetTimeline

Description

녹화된 데이터 블록의 정보를 반환합니다.

Syntax

```

void GetTimeline(
    long hTimeline,
    long nCameraIndex,
    long nTimelineIndex,
    long nType,
    long* ptStart,
    long* ptEnd,
    long* pnRecType,
)

```

```
    long* pnRecID  
);
```

Parameters

hTimeline

[in] OnSearchDay 이벤트를 통해서 얻은 타임라인 핸들.

nCameraIndex

[in] 카메라 인덱스. 0부터 시작함.

nTimelineIndex

[in] 타임라인의 인덱스. 0부터 시작함.

nType

[in] 시간 타입.

- XTIME_UTC (=0): *tStart*와 *tEnd*를 UTC 시간으로 변환함.

- XTIME_LOCAL (=1): *tStart*와 *tEnd*를 local 시간으로 변환함.

ptStart

[out] 타임라인 블록의 시작 시간.

ptEnd

[out] 타임라인 블록의 종료 시간.

pnRecType

[out] Recording type. 즉, 녹화가 시작된 원인. (표 2.10 참조) 녹화가 시작된 요인이 여러 개인 경우, bit-wise ORing한 값으로 설정됨.

pnRecID

[out] 동일 시간 구간에 2개 이상의 저장 데이터가 있을 경우 이를 구분하기 위한 ID 값.

표 2.10 Recording Type

Name	Value	Description
REC_TYPE_NONE	0x00	Not displayed
REC_TYPE_MOTION	0x01	Motion Event recording
REC_TYPE_ALARM	0x02	Sensor Event recording
REC_TYPE_VLOSS	0x04	Video loss event recording
REC_TYPE_MANUAL	0x08	Manual recording
REC_TYPE_CONTINUOUS	0x10	Schedule recording
REC_TYPE_ALL	0xFF	All types

Remarks

이 함수는 OnSearchDay, OnMotionSearch 이벤트의 이벤트 핸들러 내에서만 유효합니다.

Example

[C++]

```
// 이 함수는 OnSearchDay 이벤트가 발생했을 때 호출되는 이벤트 핸들러입니다.

void CTestSampleDlg::OnSearchDayXnssdkdevicectrl1(long nDeviceID, long nControlID, long
hTimeline)
{
    long CamCount = m_ctlXnsDevice.GetCameraCount(hTimeline);
    long TimelineCount = m_ctlXnsDevice.GetTimelineCount(hTimeline, 0);

    CString strRecInfo;
    long tStart, tEnd;
    long nRecType;
    long nRecId;
    long StartYear, StartMon, StartDay, EndYear, EndMon, EndDay;
    long StartHour, StartMin, StartSec, EndHour, EndMin, EndSec;

    OutputDebugString("<On UTC time>");
    for(int i=0; i<TimelineCount; i++)
    {
        m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
&nRecType, &nRecId);
        m_ctlXnsDevice.TimetToDate(tStart, &StartYear, &StartMon, &StartDay,
&StartHour, &StartMin, &StartSec);
        m_ctlXnsDevice.TimetToDate(tEnd, &EndYear, &EndMon, &EndDay, &EndHour,
&EndMin, &EndSec);
        // ...
    }

    OutputDebugString("<On Local time>");
    for(i=0; i<TimelineCount; i++)
    {
        m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
&nRecType, &nRecId);
        m_ctlXnsDevice.TimetToDate(tStart, &StartYear, &StartMon, &StartDay,
&StartHour, &StartMin, &StartSec);
        m_ctlXnsDevice.TimetToDate(tEnd, &EndYear, &EndMon, &EndDay, &EndHour,
&EndMin, &EndSec);
    }
}
```

```
// ...
}
}
```

See Also

[OnSearchDay](#), [GetTimelineCount](#), [TimetToDate](#), APPENDIX C [UTC Time](#)

Recording

StartLocalRecording

Description

Local PC에서 스트림 데이터 녹화를 시작합니다.

Syntax

```
long StartLocalRecording(
    long hDevice,
    long nControlID,
    LPCTSTR szFile,
    long nFileType
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

szFile

[in] 녹화된 데이터를 저장할 파일 이름.

nFileType

[in] 파일 포맷.

SEC_WRITER (1): 소형 플레이어가 포함된 비디오 스트림.

REC_WRITER (=2): 삼성 자체 포맷. 이 파일은 Samsung XNS ActiveX SDK를 이용해서만 재생이 가능함.

참고

소형 플레이어는 녹화가 종료되면, 같은 폴더에 생성됩니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_ALREADY_RUN: 이미 실행 중임.
- ERR_FILE_ALREADY_OPEN: 파일이 이미 열려있음.

Example

[C++]

```
CString strFile = "d:\test";
long nErrCode = m_ctlXnsDevice.StartLocalRecording(m_hDevice, 2, strFile, REC_WRITER);
```

See Also

[StopLocalRecording](#), [CreateFileReader](#), [ControlID](#)

StopLocalRecording

Description

Local PC에서 녹화하는 것을 중지합니다.

Syntax

```
long StopLocalRecording(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.StopLocalRecording(m_hDevice, 2);
```

See Also

[StartLocalRecording](#), [ControlID](#)

StartAviLocalRecording

Description

Local PC에서 스트림 데이터 녹화를 Avi 포맷으로 시작합니다.

Syntax

```
long StartAviLocalRecording(
    long hDevice,
    long nControlID,
    LPCTSTR szFile,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

szFile

[in] 녹화된 데이터를 저장할 파일 이름.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_ALREADY_RUN: 이미 실행 중임.
- ERR_FILE_ALREADY_OPENED: 파일이 이미 열려있음.

Example

[C++]

```
CString strFile = "d:\WWtest";
long nErrCode = m_ctlXnsDevice.StartAviLocalRecording(m_hDevice, 2, strFile);
```

See Also

[StopAviLocalRecording](#), [ControlID](#)

StopAviLocalRecording

Description

Local PC에서 Avi 포맷으로 녹화하는 것을 중지합니다.

Syntax

```
long StopLocalRecording(
    long hDevice,
    long nControlID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.StopAviLocalRecording(m_hDevice, 2);
```

See Also

[StartAviLocalRecording](#), [ControlID](#)

ToggleDvrRecording

Description

DVR/NVR에서 녹화를 시작하거나 중지합니다.

참고

이 함수는 일부 DVR/NVR 제품에서 지원되는 함수입니다. SRD-480D, SVR-9XX, SVR-16XX, SVR-3200 DVR 제품은 해당 기능을 지원하지 않습니다.

Syntax

```
long ToggleDvrRecording(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.ToggleDvrRecording(m_hDevice);
```

Backup

백업(backup) 기능은 장비에 녹화된 스트리밍 데이터를 네트워크를 통해서 Local PC에 저장하는 기능입니다.

StartBackup

Description

백업을 시작합니다.

Syntax

```
long StartBackup(
    long hDevice,
    long nControlID,
    LPCTSTR szFile,
    long nFileType,
    long tStart,
    long tEnd
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

szFile

[in] 백업 파일 이름.

nFileType

[in] 백업 파일 포맷.

값	설명	파일생성주기
SEC_WRITER = 1	소형 플레이어가 포함된 비디오 스트림	백업데이터 크기가 700MB 이상일 경우 날짜가 변경될 경우 해상도가 변경될 경우 DST 가 끝나는 시점
REC_WRITER = 2	삼성 자체 포맷. 이 파일의 경우 Samsung XNS ActiveX SDK 를 이용해서만 재생이 가능함.	1 시간 단위 (D1, 30 fps 의 경우 1 시간 백업시 데이터 크기가 600MB 임)

tStart

[in] 녹화된 데이터에서 백업을 시작할 시간. (UTC time_t 포맷)

tEnd

[in] 녹화된 데이터에서 백업을 종료할 시간. (UTC time_t 포맷)

참고

| 소형 플레이어는 백업이 종료되면, 같은 폴더에 생성됩니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.
- ERR_MEDIA_ALREADY_OPENED: 미디어가 이미 열려있음.
- ERR_FILE_ALREADY_OPENED: 파일이 이미 열려있음.
- ERR_ALREADY_RUN 이미 실행 중임.

Example

[C++]

```
CString strFile = "d:\WW\test";
long tStart = m_ctrlXnsDevice.DateToTimet(2010, 9, 6, 15, 0, 0);
long tEnd = m_ctrlXnsDevice.DateToTimet(2010, 9, 6, 20, 0, 0);

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_pParent->m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_pParent->m_hDevice, tEnd);

long errCode = m_ctrlXnsDevice.StartBackup(m_hDevice, 2, strFile, REC_WRITER, tStart, tEnd);
```

See Also

[StopBackup](#), [ControlID](#), APPENDIX C [UTC Time](#)

StopBackup

Description

백업을 중지합니다.

Syntax

```
long StopBackup(long hDevice, long nControlID);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long errCode = m_ctlXnsDevice.StopBackup(m_hDevice, 2);
```

See Also

[StartBackup](#)

StartAviBackup

Description

Avi 포맷으로 백업을 시작합니다.

Syntax

```
long StartAviBackup(
    long hDevice,
    long nControlID,
    LPCTSTR szFile,
    long tStart,
    long tEnd
    long nTrackID
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

szFile

[in] 백업 파일 이름.

tStart

[in] 녹화된 데이터에서 백업을 시작할 시간. (UTC time_t 포맷)

tEnd

[in] 녹화된 데이터에서 백업을 종료할 시간. (UTC time_t 포맷)

nTrackID

[in] 중복구간 ID.(default : -1)

참고

파일생성주기

백업데이터 크기가 1GB(default) 이상일 경우

백업 시간이 1 시간 이상일 경우

해상도가 변경될 경우

Video, Audio 코덱이 변경될 경우

Frame rate 가 변경될 경우

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.
- ERR_MEDIA_ALREADY_OPENED: 미디어가 이미 열려있음.
- ERR_FILE_ALREADY_OPENED: 파일이 이미 열려있음.
- ERR_ALREADY_RUNNING 이미 실행 중임.

Example

[C++]

```
CString strFile = "d:\test";
long tStart = m_ctrlXnsDevice.DateToTimet(2010, 9, 6, 15, 0, 0);
long tEnd = m_ctrlXnsDevice.DateToTimet(2010, 9, 6, 20, 0, 0);

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tEnd);

long errCode = m_ctrlXnsDevice.StartAviBackup(m_hDevice, 2, strFile, tUTCStart, tUTCEnd, -1);
```

See Also

[StopAviBackup](#), [ControlID](#), APPENDIX C UTC Time

StopAviBackup

Description

Avi 포맷으로 백업하는 것을 중지합니다.

Syntax

```
long StopAviBackup(long hDevice, long nControlID);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long errCode = m_ctlXnsDevice.StopAviBackup(m_hDevice, 2);
```

See Also

[StartAviBackup](#)

GetAviOptionSize

Description

현재 설정된 AVI 파일의 최대 size를 가져옵니다.

Syntax

```
long GetAviOptionSize( );
```

Return Value

현재 설정된 AVI 파일 최대 size. (1 ~ 2048 의 값을 가지며 megabyte 단위임)

Example

[C++]

```
long nAviSize = m_ctlXnsDevice.GetAviOptionSize();
```

See Also

[SetAviOptionSize](#), [StartBackup](#), [StartAviLocalRecording](#)

SetAviOptionSize

Description

AVI 파일의 최대 size를 설정합니다. 이 함수로 설정된 값은 StartAviLocalRecording, StartAviBackup 함수를 사용할 때 생성되는 Avi 파일의 최대 size 입니다.

Syntax

```
void SetAviOptionSize(long nAviSize );
```

Parameters

nAviSize

[in] Avi 파일 size. (1 ~ 2048 의 값을 가지며 megabyte 단위임)

Example

[C++]

```
m_ctlXnsDevice.SetAviOptionSize(1024);
```

See Also

[GetAviOptionSize](#), [StartBackup](#), [StartAviLocalRecording](#)

GetAviOptionPeriod

Description

현재 설정된 AVI 파일의 최대 녹화 시간을 가져옵니다.

Syntax

```
long GetAviOptionPeriod( );
```

Return Value

현재 설정된 AVI 파일 최대 녹화 시간. (10 ~ 7200 의 값을 가지며 second 단위임)

Example

[C++]

```
long nAviPeriod = m_ctlXnsDevice.GetAviOptionPeriod();
```

See Also

[SetAviOptionPeriod](#), [StartBackup](#), [StartAviLocalRecording](#)

SetAviOptionPeriod

Description

AVI 파일의 최대 녹화 시간을 설정합니다. 이 함수로 설정된 값은 StartAviLocalRecording, StartAviBackup 함수를 사용할 때 생성되는 Avi 파일의 최대 녹화 시간입니다.

Syntax

```
void SetAviOptionPeriod(long nAviPeriod);
```

Parameters

nAviPeriod

[in] Avi 파일 최대 녹화 시간. (10 ~ 7200 의 값을 가지며 second 단위임)

Example

[C++]

```
m_ctlXnsDevice.SetAviOptionPeriod(3600);
```

See Also

[GetAviOptionPeriod](#), [StartBackup](#), [StartAviLocalRecording](#)

StartSecBackup

Description

백업을 시작합니다.

Syntax

```
long StartSecBackup(
    long hDevice,
    long nControlID,
    LPCTSTR szFile,
    long nFileType,
    long tStart,
    long tEnd,
    long nTrackID,
    LPCTSTR szPassword
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nControlID

[in] Control ID.

szFile

[in] 백업 파일 이름.

nFileType

[in] 백업 파일 포맷.

값	설명	파일생성주기
SEC_WRITER = 1	소형 플레이어가 포함된 비디오 스트림	백업데이터 크기가 700MB 이상일 경우 날짜가 변경될 경우 해상도가 변경될 경우 DST 가 끝나는 시점

tStart

[in] 녹화된 데이터에서 백업을 시작할 시간. (UTC time_t 포맷)

tEnd

[in] 녹화된 데이터에서 백업을 종료할 시간. (UTC time_t 포맷)

nTrackID

[in] 중복구간 ID.(default : -1)

szPassword

[in] 백업 파일을 재생하기 위한 비밀번호.

참고

소형 플레이어는 백업이 종료되면, 같은 폴더에 생성됩니다.

Return Value

ERR_SUCCESS: 함수 수행에 성공함.

ERR_INVALID_PARAMETER: 유효하지 않은 파라미터.

ERR_NOT_CONNECT: 장비가 연결되지 않음.

ERR_NO_PERMISSION: 권한이 없음.

ERR_MEDIA_ALREADY_OPEN: 미디어가 이미 열려있음.

ERR_FILE_ALREADY_OPEN: 파일이 이미 열려있음.

ERR_ALREADY_RUN: 이미 실행 중임.

Example

[C++]

```
CString strFile = "d:\WWtest";
long tStart = m_ctlXnsDevice.DateToTimet(2010, 9, 6, 15, 0, 0);
long tEnd = m_ctlXnsDevice.DateToTimet(2010, 9, 6, 20, 0, 0);

long tUTCStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_pParent->m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_pParent->m_hDevice, tEnd);

long errCode = m_ctlXnsDevice.StartSecBackup(m_hDevice, 2, strFile, SEC_WRITER, tStart, tEnd, -1, "password");
```

See Also

[StopBackup](#), [ControlID](#), APPENDIX C [UTC Time](#)

CreateFileReader

Description

파일 재생기(file reader)를 생성합니다. 이 재생기는 백업/녹화된 파일을 재생할 때 사용합니다.

Syntax

```
long CreateFileReader(LPCTSTR szFile);
```

Parameters

szFile

[in] 재생할 파일 이름. (최대 길이는 260 바이트)

Return Value

재생기의 핸들. 이 핸들은 GetMediaSource(), GetFileName(), GetTimelineHandle(), PlayReader(), PauseReader() 등의 파라미터로 사용됩니다.

Example

[C++]

```
CString strFile = "d:\test";
long m_hFileReader = m_ctlXnsDevice.CreateFileReader(strFile);
```

See Also

[ReleaseFileReader](#), [GetMediaSource](#), [GetFileName](#), [GetTimelineHandle](#), [PlayReader](#), [PauseReader](#)

ReleaseFileReader

Description

File Reader를 해제합니다.

Syntax

```
void ReleaseFileReader(long hReader);
```

Parameters

hReader

[in] 재생기의 핸들. 이 값은 CreateFileReader()를 이용하여 얻을 수 있음.

Example

[C++]

```
// m_hFileReader는 CreateFileReader()에서 리턴되는 핸들입니다.
m_ctlXnsDevice.ReleaseFileReader(m_hFileReader);
```

See Also

[CreateFileReader](#)

GetFileName

Description

주어진 파일 핸들의 파일 이름과 경로를 반환합니다.

Syntax

```
CString GetFileName(long hReader);
```

Parameters

hReader

[in] 재생할 파일의 핸들. 이 값은 CreateFileReader()를 이용하여 얻을 수 있음.

Return Value

*hReader*가 가리키는 파일의 이름과 경로.

Example

[C++]

```
// m_hFileReader는 CreateFileReader()에서 리턴되는 핸들입니다.
```

```
CString filename = m_ctlXnsDevice.GetFileName(m_hFileReader);
```

See Also

[CreateFileReader](#)

GetMediaSource

Description

파일 핸들로부터 미디어 소스의 핸들을 가져옵니다. 이 값은 XnsSdkWindows를 이용하여 이미지를 표시할 때 사용됩니다.

Syntax

```
long GetMediaSource(long hReader);
```

Parameters

hReader

[in] 재생할 파일의 핸들. 이 값은 CreateFileReader()를 이용하여 얻을 수 있음.

Return Value

미디어 소스의 핸들.

Example

[C++]

// m_hFileReader는 CreateFileReader()에서 리턴되는 핸들입니다.

```
long hMediaSource = m_ctlXnsDevice.GetMediaSource(m_hReader);
```

See Also

[CreateFileReader](#)

GetTimelineHandle

Description

파일 핸들로부터 타임라인 핸들을 가져옵니다. 이 값은 녹화된 구간의 영상을 화면에 표시하기 위해 사용됩니다.

Syntax

```
long GetTimelineHandle(long hReader);
```

Parameters

hReader

[in] 재생할 파일의 핸들. 이 값은 CreateFileReader()를 이용하여 얻을 수 있음.

Return Value

타임라인 핸들.

Example

[C++]

// m_hFileReader는 CreateFileReader()에서 리턴되는 핸들입니다.

```
long hTimeline = m_ctlXnsDevice.GetTimelineHandle(m_hReader);
```

See Also

[CreateFileReader](#)

PlayReader

Description

백업된 파일을 지정된 속도로 재생합니다.

Syntax

```
long PlayReader(
    long hReader,
    double dSpeed
);
```

Parameters

hReader

[in] 재생기의 핸들. 이 값은 CreateFileReader()를 이용하여 얻을 수 있음.

dSpeed

[in] 재생 속도(배속).

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
// m_hFileReader는 CreateFileReader()에서 리턴되는 핸들입니다.  
long nErrCode = m_ctlXnsDevice.PlayReader(m_hFileReader, 1);
```

See Also

[CreateFileReader](#), [PauseReader](#)

PauseReader

Description

백업된 파일 재생을 일시 중지 합니다. PlayReader()를 이용하여 재생을 계속할 수 있습니다.

Syntax

```
long PauseReader(long hReader);
```

Parameters

hReader

[in] 재생기의 핸들. 이 값은 CreateFileReader()를 이용하여 얻을 수 있음.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
// m_hFileReader는 CreateFileReader()에서 리턴되는 핸들입니다.  
long nErrCode = m_ctlXnsDevice.PauseReader(h.FileReader);
```

See Also

[CreateFileReader](#), [PlayReader](#)

ConvertRec2Avi

Description

REC 파일을 AVI 파일로 변환합니다.

Syntax

```
long ConvertRec2Avi(LPCTSTR szFile);
```

Parameters

szFile

[in] REC 파일 이름. (최대 길이는 260 바이트)

Return Value

파일 핸들. 이 핸들은 StopRec2Avi()의 파라미터로 사용됩니다.

Example

[C++]

```
CString strFile = "d:\WWtest";  
long m_hFileHandle = m_ctlXnsDevice.ConvertRec2Avi(strFile);
```

See Also

[StopRec2Avi](#)

StopRec2Avi

Description

REC 파일을 AVI 파일로 변환을 중지합니다.

Syntax

```
void StopRec2Avi(long hFileHandle);
```

Parameters

hFileHandle

[in] 파일의 핸들. 이 값은 ConvertRec2Avi()를 이용하여 얻을 수 있음.

Example

[C++]

```
// hFileHandle은 ConvertRec2Avi()에서 리턴되는 핸들입니다.
```

```
m_ctlXnsDevice.StopRec2Avi(hFileHandle);
```

See Also

[ConvertRec2Avi](#)

Update Firmware and Import/Export Configuration

UploadSoftware

Description

장비의 펌웨어(firmware)를 업그레이드 하기 위해 펌웨어 파일을 업로드(upload)합니다. 업로드

의 진행/완료 상황은 OnUploadSoftwareProgress 이벤트를 통해 전달됩니다. 업로드가 실패한 경우, OnUploadSoftwareProgress 이벤트가 발생합니다.

Syntax

```
long UploadSoftware(
    long hDevice,
    LPCTSTR szFile
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

szFile

[in] 펌웨어 파일 이름. (최대길이는 1024바이트임)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
// m_hDevice는 CreateDevice()에서 리턴되는 핸들입니다.
long nErrCode = m_ctlXnsDevice.UploadSoftware(m_hDevice, "d:\shr-detroit-
pkg_v2.10_100723093736.img");
DisplayErrCode("OnUploadSoftware", nErrCode);
```

See Also

[CreateDevice](#), [OnUploadSoftwareProgress](#)

ImportConfig

Description

설정 파일(configuration file)을 장비에 전송 및 적용하는 함수입니다. 설정 파일을 장비로 내보내는 진행/완료 상황은 OnImportConfigProgress 이벤트를 통해 전달됩니다.

실패할 경우에는, OnImportConfigProgress 이벤트 핸들러내의 nErrorCode는 0(ERR_SUCCESS)

과 84(FILE_TRANSFERRING)가 아닌 값이 전달됩니다. 2011.7월 현재 이 함수는 DVR 디바이스에 한해서 지원합니다.

Syntax

```
long ImportConfig(
    long hDevice,
    LPCTSTR szFile,
    BOOL bWithNetSetting
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

szFile

[in] 임포트할 설정 파일 이름.

bWithNetSetting

[in] 설정 정보가 네트워크 설정 정보를 포함하고 있는지를 나타냄.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
// m_hDevice는 CreateDevice()에서 리턴되는 핸들입니다.
long nErrCode = m_ctlXnsDevice.ImportConfig(m_hDevice, "d:WWWdvrConfig", FALSE);
DisplayErrCode("OnImportConfig", nErrCode);
```

See Also

[CreateDevice](#), [ExportConfig](#)

ExportConfig

Description

장비로부터 설정 파일(configuration file)을 가져옵니다. 설정 파일을 가져오는 진행/완료 상황

은OnExportConfigProgress 이벤트를 통해서 전달됩니다.

실패할 경우에는, OnExportConfigProgress 이벤트 핸들러내의 nErrorCode는 0(ERR_SUCCESS)과 84(FILE_TRANSFERRING)가 아닌 값이 전달됩니다. 2011.7월 현재 이 함수는 DVR 디바이스에 대해서 지원합니다.

Syntax

```
long ExportConfig(
    long hDevice,
    LPCTSTR szFile
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

szFile

[in] 장비로 보낼 설정 파일 이름.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
// m_hDevice는 CreateDevice()에서 리턴되는 핸들입니다.
long nErrCode = m_ctlXnsDevice.ExportConfig(m_hDevice, "d:\WWConfig");
DisplayErrCode("OnExportConfig", nErrCode);
```

See Also

[ImportConfig](#)

Device Profile

LoadDeviceProfile

Description

device.xml 파일에서 장비 정보를 읽어옵니다. device.xml 파일은 다음과 같은 정보를 갖고 있습니다.

표 2.11 Device Information in device.xml

Field	Description
VENDORNAME	String. Max 127 bytes
MODELNAME	String. Max 127 bytes
ADDRESSTYPE	Address type to connect to the device.
ADDRESS	String. Max 255 bytes. Address to connect to the device.
PORT	Port number to connect to the device
HTTPPORT	Http port number to connect to the device
USERID	User ID to log-in to the device
PASSWORD	Password to log-in to the device
LATESTUPDATE	Latest update time. time_t format
CONTROLNAME	Control Names and Devices name.

Syntax

```
long LoadDeviceProfile(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_CFG_FILE: 잘못된 Xns.xml 파일.

Example

[C++]

```
// m_hDevice는 CreateDevice()에서 리턴되는 핸들입니다.  
long nErrCode = m_ctlXnsDevice.LoadDeviceProfile(m_hDevice);  
DisplayErrCode("LoadDeviceProfile ", nErrCode);
```

See Also

[CreateDevice](#), [SaveDeviceProfile](#), [DeleteDeviceProfile](#)

SaveDeviceProfile

Description

장비 정보를 device.xml에 저장합니다. device.xml에 저장되는 정보는 표3.10를 참조하시기 바랍니다.

Syntax

```
long SaveDeviceProfile(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_CFG_FILE: 잘못된 Xns.xml 파일.

Example

[C++]

```
// m_hDevice는 CreateDevice()에서 리턴되는 핸들입니다.  
long nErrCode = m_ctlXnsDevice.SaveDeviceProfile(m_hDevice);  
DisplayErrCode("SaveDeviceProfile ", nErrCode);
```

See Also

[CreateDevice](#), [LoadDeviceProfile](#), [DeleteDeviceProfile](#)

DeleteDeviceProfile

Description

device.xml 파일에서 장비 정보를 삭제합니다.

Syntax

```
long DeleteDeviceProfile(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_INVALID_CFG_FILE: 잘못된 Xns.xml 파일.

Example

[C++]

```
// m_hDevice는 CreateDevice()에서 리턴되는 핸들입니다.  
long nErrCode = m_ctlXnsDevice.DeleteDeviceProfile(m_hDevice);  
DisplayErrCode("DeleteDeviceProfile ", nErrCode);
```

See Also

[CreateDevice](#), [LoadDeviceProfile](#), [SaveDeviceProfile](#)

XML File

ReadConfigValue

Description

config.xml 파일에 저장된 설정(configuration) 값을 읽어옵니다.

Syntax

```
CString ReadConfigValue(LPCTSTR szKey);
```

Parameters

szKey

[in] 읽어오고자 하는 설정 항목.

Return Value

*szKey*에 대응하는 설정 값. 만약, 설정 파일이 존재하지 않는다면, NULL이 리턴됩니다. 설정 값이 존재하지 않는 항목이라면(예, <DATETIME/>), ""이 리턴됩니다.

Example

[C++]

```
CString strValue = m_ctlXnsDevice.ReadConfigValue("aspectratio");
```

See Also

[WriteConfigValue](#), [DeleteConfigValue](#)

WriteConfigValue

Description

config.xml 파일에 설정 정보를 저장합니다.

Syntax

```
BOOL WriteConfigValue(
    LPCTSTR szKey,
    LPCTSTR szValue
);
```

Parameters

szKey

[in] 설정 항목.

szValue

[in] 설정 항목에 대응하는 설정 값.

Return Value

이 함수가 성공적으로 수행되면 TRUE를 리턴합니다. 다음과 같은 경우, FALSE가 리턴됩니다.

- config.xml이 없는 경우
- XROOT 항목이 없는 경우
- 항목(key) 추가에 실패한 경우
- 값(value) 설정에 실패한 경우

Example

[C++]

```
BOOL bSave = m_ctlXnsDevice.WriteConfigValue("aspectratio", "true");
```

See Also

[ReadConfigValue](#), [DeleteConfigValue](#)

DeleteConfigValue

Description

config.xml 파일에서 특정 설정 정보를 삭제합니다.

Syntax

```
void DeleteConfigValue(LPCTSTR szKey);
```

Parameters

szKey

[in] 삭제할 설정 항목.

Example

[C++]

```
m_ctlXnsDevice.DeleteConfigValue("aspectratio");
```

See Also

[ReadConfigValue](#), [WriteConfigValue](#)

CopyXml

Description

config.xml 파일의 특정 항목을 다른 파일에 복사합니다.

Syntax

```
BOOL CopyXml(  
    LPCTSTR szFile1,  
    LPCTSTR szKey1,
```

```
LPCTSTR szFile2,
LPCTSTR szKey2
);
```

Parameters

szFile1

[in] 복사할 파일 이름. (Source file)

szKey1

[in] 복사할 설정 항목.

szFile2

[in] 복사된 파일 이름. (Destination file)

szKey2

[in] szFile2에 복사된 설정 항목

Return Value

이 함수가 성공적으로 수행되면 TRUE(=1)가 리턴됩니다. 실패하면 FALSE(=0)가 리턴됩니다.

Example

[C++]

```
BOOL bSuccess = m_ctlXnsDevice.CopyXml("./config.xml", "aspectratio", "./config2.xml",
"aspectratio");
```

FindConfigChildeKeyList

Description

특정 항목(key)로부터 child list를 얻어옵니다.

Syntax

```
long FindConfigChildeKeyList(LPCTSTR szKey);
```

Parameters

szKey

[in] 검색할 설정 항목. (최대 길이 63 바이트)

Return Value

검색 결과에 대한 핸들러.

Example

[C++]

```
Long hFind = m_ctlXnsDevice.FindConfigChildeKeyList("ROOT");
```

Miscellaneous

DateToTimet

Description

날짜(년/월/일)와 시간(시/분/초)를 time_t 포맷으로 변환합니다.

Syntax

```
long DateToTimet(
    long nYear,
    long nMonth,
    long nDay,
    long nHour,
    long nMinute,
    long nSecond
);
```

Parameters

nYear

[in] 년(year)

nMonth

[in] 월(month)

nDay

[in] 일(day)

nHour

[in] 시|(hour)

nMinute

[in] 분(minute)

nSecond

[in] 초(second)

Return Value

time_t 포맷으로 변환한 값.

Example

[C++]

```
long tPos = m_ctlXnsDevice.DateToTimet(2010, 9, 6, 15, 0, 0);
```

See Also

[TimetToDate](#)

TimetToDate

Description

time_t 포맷의 데이터를 날짜(년/월/일)와 시간(시/분/초)으로 변환합니다.

Syntax

```
void TimetToDate(
    long tTime,
    long* pnYear,
    long* pnMonth,
    long* pnDay,
    long* pnHour,
    long* pnMinute,
    long* pnSecond
);
```

Parameters

tTime

[in] 변환할 time_t 포맷의 데이터.

pnYear

[out] 년(year)

pnMonth

[out] 월(month)

pnDay

[out] 일(day)

pnHour

[out] 시(hour)

pnMinute

[out] 분(minute)

pnSecond

[out] 초(second)

Example

[C++]

// 이 함수는 OnSearchDay 이벤트가 발생했을 때 호출되는 이벤트 핸들러입니다.

```
void CTestSampleDlg::OnSearchDayXnssdkdevicectrl1(long nDeviceID, long nControlID, long hTimeline)
{
    // ...
    long TimelineCount = m_ctlXnsDevice.GetTimelineCount(hTimeline, 0);

    CString strRecInfo;
    long tStart, tEnd;
    long nRecType;
    long nRecId;
    long StartYear, StartMon, StartDay, EndYear, EndMon, EndDay;
    long StartHour, StartMin, StartSec, EndHour, EndMin, EndSec;

    for(int i=0; i<TimelineCount; i++)
    {
        m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
        &nRecType, &nRecId);
        m_ctlXnsDevice.TimetToDate(tStart, &StartYear, &StartMon, &StartDay,
        &StartHour, &StartMin, &StartSec);
        m_ctlXnsDevice.TimetToDate(tEnd, &EndYear, &EndMon, &EndDay, &EndHour,
        &EndMin, &EndSec);
        // ...
    }
}
```

See Also

[DateToTimet](#), [ControlID](#)

LocalTimeToDeviceUTC

Description

Local Time 을 UTC Time 으로 변환합니다.

Syntax

```
long LocalTimeToDeviceUTC(
    long hDevice,
    long tLocal,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

tLocal

[in] local time_t 값

Return Value

UTC Time으로 변환한 time_t 값.

Note

UTC(Coordinated Universal Time) 협정시계시는 고정밀도 원자 시간 표준이다. 전 세계의 표준 시간대는 UTC에서의 양 또는 음의 오프셋으로 표시된다. 따라서 UTC 시간을 통해 표준 시간 대에 대해 종립적인 시간을 제공한다. UTC 00:00은 그리니치 표준시이다. 한국의 경우 UTC+9:00 이므로 한국의 오전11:00은 UTC Time 변환시 9시간을 뺀 오전 3:00시이고 반대로 UTC Time 오전 11:00시를 Local Time으로 변환하면 오후 8:00 시이다.

Example

[C++]

```
long tUTCTStart = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tStart);
long tUTCEnd = m_ctrlXnsSdkDevice.LocalTimeToDeviceUTC(m_hDevice, tEnd);
```

See Also

[DateToTimet](#), [TimetToDate](#), APPENDIX C. [UTC Time](#)

UTCToDeviceLocalTime

Description

UTC Time을 Local Time으로 변환합니다.

Syntax

```
long UTCToDeviceLocalTime (
    long hDevice,
    long tUTC,
);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

tUTC

[in] UTC time_t 값

Return Value

Local Time으로 변환한 time_t 값.

Note

UTC(Coordinated Universal Time) 협정시계시는 고정밀도 원자 시간 표준이다. 전 세계의 표준 시간대는 UTC에서의 양 또는 음의 오프셋으로 표시된다. 따라서 UTC 시간을 통해 표준 시간 대에 대해 중립적인 시간을 제공한다. UTC 00:00은 그리니치 표준시이다. 한국의 경우 UTC+9:00 이므로 한국의 오전11:00은 UTC Time 변환시 9시간을 뺀 오전 3:00시이고 반대로 UTC Time 오전 11:00시를 Local Time으로 변환하면 오후 8:00 시이다.

Example

[C++]

```
tLocalStart = m_ctrlXnsSdkDevice.UTCToDeviceLocalTime(m_hDevice, tStart);
tLocalEnd = m_ctrlXnsSdkDevice.UTCToDeviceLocalTime(m_hDevice, tEnd);
```

See Also

[DateToTimet](#), [TimetToDate](#), APPENDIX C. [UTC Time](#)

AutoScan

Description

네트워크에 연결된 모든 장비를 검색하도록 요청하는 함수입니다. 이 함수의 수행 결과는 OnDeviceDetected, OnDeviceDetected2 이벤트를 통해서 전달됩니다.

Syntax

```
long AutoScan(long nScanDuration);
```

Parameters

nScanDuration

[in] 장비 정보를 수신하기 위한 대기 시간. (단위: 초)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.AutoScan(10);
```

See Also

[IPInstall](#), [OnDeviceDetected](#), [OnDeviceDetected2](#), [OnDeviceDetected3](#)

IPInstall

Description

네트워크 카메라의 네트워크 설정(network configuration)을 설정합니다. 이 함수는 네트워크 카메라에서만 유효하며 수행 결과는 OnDeviceInstalled 이벤트를 통해서 전달됩니다.

Syntax

```
long IPInstall(
    LPCTSTR szVendorName,
    LPCTSTR szmodelName,
    LPCTSTR szIP,
    LPCTSTR szSubnetMask,
    LPCTSTR szGateway,
    LPCTSTR szMac,
```

```

        long nPort,
        long nHttpPort,
        LPCTSTR szPassword
);

```

Parameters

szVendorName

[in] 벤더명. 'Samsung'으로 고정됨.

szModelName

[in] 모델명. 현재 사용하지 않음.

szIP

[in] IP address. (최대 길이 256 바이트)

szSubnetMask

[in] Subnet mask. (최대 길이 256 바이트)

szGateway

[in] Gateway. (최대 길이 256 바이트)

szMac

[in] MAC address. (12글자로 고정됨.)

nPort

[in] 장비의 포트 번호.

nHttpPort

[in] HTTP 포트 번호.

szPassword

[in] 관리자(admin) 권한으로 로그인하기 위한 비밀번호.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.

Example

[C++]

```

long nErrCode = m_ctlXnsDevice.IpInstall("Samsung", "SNB-5000", "192.168.1.200",
"255.255.255.0", "192.168.1.1", "00166cabcd", 4520, 80, "4321");

```

See Also

[AutoScan](#), [OnDeviceInstalled](#)

IPIInstallDhcp

Description

IPIInstall과 같은 역할을 하는 함수로 네트워크 설정(network configuration)을 DHCP로 설정합니다.

Syntax

```
long IPIInstallDhcp(
    LPCTSTR szVendorName,
    LPCTSTR szmodelName,
    LPCTSTR szMac,
    long nPort,
    long nHttpPort,
    LPCTSTR szPassword
);
```

Parameters

szVendorName

[in] 벤더명. 'Samsung'으로 고정됨.

szmodelName

[in] 모델명. 현재 사용하지 않음.

szMac

[in] MAC address. (12글자로 고정됨.)

nPort

[in] 장비의 포트 번호.

nHttpPort

[in] HTTP 포트 번호.

szPassword

[in] 관리자(admin) 권한으로 로그인하기 위한 비밀번호.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.IpInstallDhcp("Samsung", "SNB-5000", "00166abcdef", 4520,
80, "4321");
```

See Also

[AutoScan](#), [OnDeviceInstalled](#)

SetEventDiscardTime

Description

이벤트가 삭제되는 시간(event discard time)을 설정합니다. 이 시간 안에 동일한 이벤트가 발생하면, 그 이벤트는 삭제됩니다.

Syntax

```
long SetEventDiscardTime(
    long nEventType,
    long nDiscardTime
);
```

Parameters*nEventType*

[in] 이벤트 타입

- EVENT_ALL (=0xff): 모든 이벤트 타입에 적용됨.
- EVENT_ALARM (=0x01): 알람 이벤트에 적용됨.
- EVENT_MOTION (=0x02): 모션(motion) 이벤트에 적용됨.
- EVENT_VIDEOLOSS (=0x04): 영상 손실(video loss) 이벤트에 적용됨.

nDiscardTime

[in] Event discard time. (단위: 초)

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.

Example**[C++]**

```
long nErrCode = m_ctlXnsDevice.SetEventDiscardTime(30);
```

GetErrorString

Description

에러 코드에 대한 설명을 반환합니다.

Syntax

```
BSTR GetErrorString(long nErrorCode);
```

Parameters

nErrorCode

[in] 에러 코드.

Return Value

에러 코드에 대한 설명. 예를 들어, *nErrorCode*가 0이라면, 애플리케이션은 "ERR_SUCCESS"라는 문자열을 리턴 값으로 얻을 수 있고, *nErrorCode*가 0x10이라면, "ERR_INVALID_CFG_FILE"이라는 문자열을 리턴 값으로 받습니다.

Example

[C++]

```
if (nErrCode)
{
    OutputDebugString(m_ctlXnsDevice.GetErrorString(nErrCode));
}
```

See Also

APPENDIX A. DEP

문제

PowerRestart

Description

장비 재부팅을 수행합니다. 이 함수의 결과는 OnPowerRestart 이벤트를 통해서 받을 수 있습니다.

니다.

Syntax

```
long PowerRestart(long hDevice);
```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_CONNECT: 장비가 연결되지 않음.
- ERR_NO_PERMISSION: 권한이 없음.

Example

[C++]

```
long nErrCode = m_ctlXnsDevice.PowerRestart(m_hDevice);
```

Events

이 절에서는 장비에서 발생할 수 있는 이벤트 또는 XnsSdkDevice에 정의된 함수를 호출했을 때 발생하는 이벤트에 대해 설명합니다. 아래 나열된 함수들은 해당 이벤트가 발생했을 때, 호출되는 이벤트 핸들러입니다.

OnControlLocalNameChanged

Description

로컬 PC에서 control module의 이름이 변경되면 ActiveX object에서 OnControlLocalNameChanged 이벤트를 발생시킵니다. 또한 이 이벤트는 애플리케이션이 ChangeControlLocalName()을 이용하여 control module의 이름을 변경했을 때도 발생합니다.

Syntax

```
void OnControlLocalNameChanged(
    long nDeviceID,
    long nControlID
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

Example

[C++]

```
void CTestSampleDlg::OnControlLocalNameChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID)
{
    OutputDebugString("Event Local Name Changed\n");
    char str[300];

    int hDevice = m_ctlXnsDevice.GetDeviceHandle(nDeviceID);

    // 변경된 control module 이름은 GetControlLocalName()을 이용하여 확인할 수 있습니다.
    sprintf(str, "nDeviceID[%d], nControlID[%d], Changed Name[%s]\n",
nDeviceID, nControlID, m_ctlXnsDevice.GetControlLocalName(hDevice, nControlID));
    OutputDebugString(str);
}
```

See Also

[ChangeControlLocalName](#), [GetControlLocalName](#), [ControlID](#)

OnDeviceDetected

Description

이 이벤트는 애플리케이션이 AutoScan()을 이용하여 스캐닝 메시지(scanning message)를 브로드캐스팅(broadcasting)하고 장비로부터 응답을 수신하면 발생합니다. 이때, 각각의 응답이 올 때마다 이 이벤트가 발생합니다.

Syntax

```
void OnDeviceDetected(
    BSTR szVendorName,
    BSTR szModelName,
    BSTR szDeviceName,
    BSTR szIP,
    BSTR szSubnetMask,
    BSTR szGateway,
    BSTR szMac,
    long nPort,
    long nHttpPort,
    BOOL bSupportIpInstall
);
```

Parameters

szVendorName

[in] 벤더명. (예, SAMSUNG)

szmodelName

[in] 모델명. (예, SHR-604X)

szDeviceName

[in] 장비명. (예, SHR-604X)

szIP

[in] IP 주소. (예, 192.168.10.100)

szSubnetMask

[in] 서브넷 마스크 (예, 255.255.255.0)

szGateway

[in] 게이트웨이 (예, 192.168.10.1)

szMac

[in] MAC address (예, 000918D000A3)

nPort

[in] 장비의 포트 번호 (예, 554)

nHttpPort

[in] 장비의 HTTP 포트 번호 (예, 80)

bSupportIpInstall

[in] 장비에서 IP 설치 기능(IP Install capability)을 제공하는지 여부를 나타냄.

- True (1): 지원함.

- False (0): 지원하지 않음.

Example

[C++]

```
void CTestSampleDlg::OnDeviceDetectedXnssdkdevicectrl(LPCTSTR szVendorName, LPCTSTR
sz modelName, LPCTSTR szDeviceName, LPCTSTR szIP, LPCTSTR szSubnetMask, LPCTSTR
szGateway, LPCTSTR szMac, long nPort, long nHttpPort, BOOL bSupportIpInstall)
{
    char str[300];
    sprintf(str, "Vendor[%s], Model[%s], Device[%s], ip[%s], subnet[%s], Gateway[%s],
Mac[%s], Port[%d], HttpPort[%d], bSupportIpInstall[%d]\n", szVendorName, szmodelName,
szDeviceName, szIP, szSubnetMask, szGateway, szMac, nPort, nHttpPort, bSupportIpInstall);

    OutputDebugString(str);
}
```

See Also

[AutoScan](#)

OnDeviceDetected2

Description

이 이벤트는 OnDeviceDetected()와 같은 역할을 하는 이벤트이며 DHCP 모드 설정 여부를 확인할 수 있습니다.

Syntax

```
void OnDeviceDetected2(
    BSTR szVendorName,
    BSTR szmodelName,
    BSTR szDeviceName,
    BSTR szIP,
    BSTR szSubnetMask,
    BSTR szGateway,
    BSTR szMac,
    long nPort,
    long nHttpPort,
    BSTR szModelType,
```

```
    BOOL bSupportIpInstall,
    long nNetworkMode
);
```

Parameters

szVendorName

[in] 벤더명. (예, SAMSUNG)

szModelName

[in] 모델명. (예, SHR-604X)

szDeviceName

[in] 장비명. (예, SHR-604X)

szIP

[in] IP 주소. (예, 192.168.10.100)

szSubnetMask

[in] 서브넷 마스크 (예, 255.255.255.0)

szGateway

[in] 게이트웨이 (예, 192.168.10.1)

szMac

[in] MAC address (예, 000918D000A3)

nPort

[in] 장비의 포트 번호 (예, 554)

nHttpPort

[in] 장비의 HTTP 포트 번호 (예, 80)

szModelType

[in] 장비의 모델 타입. (예, Samsung Network Camera/Encoder)

bSupportIpInstall

[in] 장비에서 IP 설치 기능(IP Install capability)을 제공하는지 여부를 나타냄.

- True (1): 지원함.
- False (0): 지원하지 않음.

nNetworkMode

[in] 장비의 DHCP 모드 설정 여부를 나타냄.

- True (1): DHCP 모드.
- False (0): STATIC 모드.

Example

[C++]

```
void CTestSampleDlg::OnDeviceDetected2Xnssdkdevicectrl(LPCTSTR szVendorName, LPCTSTR
szModelName, LPCTSTR szDeviceName, LPCTSTR szIP, LPCTSTR szSubnetMask, LPCTSTR
szGateway, LPCTSTR szMac, long nPort, long nHttpPort, LPCTSTR szModelType, BOOL
bSupportIpInstall, long nNetworkMode)
{
    char str[300];
    sprintf(str, "Vendor[%s], Model[%s], Device[%s], ip[%s], subnet[%s], Gateway[%s],
Mac[%s], Port[%d], HttpPort[%d], ModelType[%s], SupportIpInstall[%d], NetworkMode[%d]\n",
szVendorName, szModelName, szDeviceName, szIP, szSubnetMask, szGateway, szMac, nPort,
nHttpPort, szModelType, bSupportIpInstall, nNetworkMode);

    OutputDebugString(str);
}
```

See Also

[AutoScan](#)

OnDeviceDetected3

Description

이 이벤트는 OnDeviceDetected()와 같은 역할을 하는 이벤트이며 장비의 프로토콜 탑입을 확인할 수 있습니다.

Syntax

```
void OnDeviceDetected3(
    BSTR szVendorName,
    BSTR szModelName,
    BSTR szDeviceName,
    BSTR szIP,
    BSTR szSubnetMask,
    BSTR szGateway,
    BSTR szMac,
    long nPort,
    long nHttpPort,
    BSTR szModelType,
    BOOL bSupportIpInstall,
```

```

        long nNetworkMode,
        BOOL bIsDhcpIPInstall,
        long nDeviceProtocolType,
        LPCTSTR szUUID
);

```

Parameters

szVendorName

[in] 벤더명. (예, SAMSUNG)

szModelName

[in] 모델명. (예, SHR-604X)

szDeviceName

[in] 장비명. (예, SHR-604X)

szIP

[in] IP 주소. (예, 192.168.10.100)

szSubnetMask

[in] 서브넷 마스크 (예, 255.255.255.0)

szGateway

[in] 게이트웨이 (예, 192.168.10.1)

szMac

[in] MAC address (예, 000918D000A3)

nPort

[in] 장비의 포트 번호 (예, 554)

nHttpPort

[in] 장비의 HTTP 포트 번호 (예, 80)

szModelType

[in] 장비의 모델 타입. (예, Samsung Network Camera/Encoder)

bSupportIpInstall

[in] 장비에서 IP 설치 기능(IP Install capability)을 제공하는지 여부를 나타냄.

True (1): 지원함.

False (0): 지원하지 않음.

nNetworkMode

[in] 장비의 DHCP 모드 설정 여부를 나타냄.

True (1): DHCP 모드.

False (0): STATIC 모드.

nbIsDhcpIPInstall

[in] DHCP로 IP 변경 가능 여부.

True (1): 가능함.

False (0): 변경 불가.

nDeviceProtocolType

[in] 장비의 프로토콜 타입.

XDEVICE_PROTOCOL_UNKNOWN(0) : UNKNOWN.

XDEVICE_PROTOCOL_SUNAPI(1) : SUNAPI.

szUUID

[in] 사용 안함.

Example**[C++]**

```
void CTestSampleDlg::OnDeviceDetected3Xnssdkdevicectrl(LPCTSTR szVendorName, LPCTSTR
szmodelName, LPCTSTR szDeviceName, LPCTSTR szIP, LPCTSTR szSubnetMask, LPCTSTR
szGateway, LPCTSTR szMac, long nPort, long nHttpPort, LPCTSTR szModelType, BOOL
bSupportIpInstall, long nNetworkMode, BOOL nbIsDhcpIPInstall, long nDeviceProtocolType,
LPCTSTR szUUID)
{
    char str[300];
    sprintf(str, "Vendor[%s], Model[%s], Device[%s], ip[%s], subnet[%s], Gateway[%s],
    Mac[%s], Port[%d], HttpPort[%d], ModelType[%s], SupportIpInstall[%d], NetworkMode[%d],
    DhcpIPInstall[%d], ProtocolType[%d], szUUID[%s]\n", szVendorName, szmodelName,
    szDeviceName, szIP, szSubnetMask, szGateway, szMac, nPort, nHttpPort, szModelType,
    bSupportIpInstall, nNetworkMode, nbIsDhcpIPInstall, nDeviceProtocolType, szUUID);

    OutputDebugString(str);
}
```

See Also

[AutoScan](#)

OnDeviceInstalled**Description**

이 이벤트는 애플리케이션이 IPInstall(), IPInstallDhcp를 호출했을 때 발생합니다.

Syntax

```
void OnDeviceInstalled(
    long nErrorCode,
    BSTR szVendorName,
    BSTR sz modelName,
    BSTR szDeviceName,
    BSTR szIP,
    BSTR szSubnetMask,
    BSTR szGateway,
    BSTR szMac,
    long nPort,
    long nHttpPort,
    BSTR szPassword,
    long nNetworkMode
);
```

Parameters

nErrorCode

[in] 에러코드.

szVendorName

[in] 벤더명. (예, SAMSUNG)

szmodelName

[in] 모델명. (예, SHR-604X)

szDeviceName

[in] 장비명. (예, SHR-604X)

szIP

[in] IP 주소. (예, 192.168.10.100)

szSubnetMask

[in] 서브넷 마스크 (예, 255.255.255.0)

szGateway

[in] 게이트웨이 (예, 192.168.10.1)

szMac

[in] MAC address (예, 000918D000A3)

nPort

[in] 장비의 포트 번호 (예, 554)

nHttpPort

[in] 장비의 HTTP 포트 번호 (예, 80)

szPassword

[in] 관리자(admin) 권한으로 로그인하기 위한 비밀번호.

nNetworkMode

[in] 장비의 DHCP 모드 설정 여부를 나타냄.

- True (1): DHCP 모드.

- False (0): STATIC 모드.

Example

[C++]

```
void CTestSampleDlg::OnDeviceInstalledXnssdkdevicectrl(long nErrorCode, LPCTSTR
szVendorName, LPCTSTR szModelName, LPCTSTR szDeviceName, LPCTSTR szIP, LPCTSTR
szSubnetMask, LPCTSTR szGateway, LPCTSTR szMac, long nPort, long nHttpPort, LPCTSTR
szPassword, long nNetworkMode)
{
    char str[300];
    sprintf(str, "Error[%s], Vendor[%s], Model[%s], Device[%s], ip[%s], subnet[%s],
Gateway[%s], Mac[%s], Port[%d], HttpPort[%d], Password[%s], NetworkMode[%d]\n",
m_ctlXnsDevice.GetErrorString(nErrorCode), szVendorName, szModelName, szDeviceName, szIP,
szSubnetMask, szGateway, szMac, nPort, nHttpPort, szPassword, nNetworkMode);

    OutputDebugString(str);
}
```

See Also

[IPInstall](#), [IPInstallDhcp](#)

OnConnectFailed

Description

애플리케이션이 ConnectNonBlock()를 이용하여 Non-block connection을 시도했을 때, 이 작업이 실패하면 OnConnectFailed 이벤트가 발생합니다. Connect()는 함수 수행이 실패했을 때, 곧 바로 에러를 리턴하기 때문에, 이 이벤트가 발생하지는 않습니다.

Syntax

```
void OnConnectFailed(
```

```
    long nDeviceID,
    long nErrorCode
);
```

Parameters

nDeviceID

[in] 장비 ID.

nErrorCode

[in] 에러 코드

Example

[C++]

```
void CTestSampleDlg::OnConnectFailedXnssdkdevicectrl1(long nDeviceID, long nErrorCode)
{
    char str[100];
    sprintf(str, "Connection Fail.. nDeviceID[%d], Error[%s]\n", nDeviceID,
m_ctlXnsDevice.GetErrorString(nErrorCode));

    OutputDebugString(str);
}
```

See Also

[ConnectNonBlock](#), [Connect](#)

OnSearchCalendar

Description

이 이벤트는 애플리케이션이 SearchCalendar()를 호출했을 때 발생하며, 검색 결과를 포함하고 있습니다.

Syntax

```
void OnSearchCalendar(
    long nDeviceID,
    long nControlID,
    long nRecDay
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nRecDay

[in] SearchCalendar()에서 지정한 달에서 녹화된 영상이 있는 날들을 반환합니다. nRecDay의 각 비트는 그 달의 날짜를 나타내며, 가장 우측의 비트(the most-right bit)가 1일입니다. 한 달은 최대 31일까지 있기 때문에, 가장 좌측의 비트(the most-left bit, 32번째 비트)는 0이 되어야 합니다.

예를 들어, nRecDay가 "0001 0000 0000 0000 0000 0000 0000 1011"의 비트 조합을 갖는다면, 1일, 2일, 4일, 29일에 녹화된 영상이 있다는 뜻입니다.

Example

[C++]

```
void CTestSampleDlg::OnSearchTextCalendarXnssdkdevicectrl1(long nDeviceID, long nControlID,
long nRecDay)
{
    CString strDay = "01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31\r\n";
    OutputDebugString(strDay);

    long nBitFlag = 0x01;
    CString strRec;

    for (int i=0; i<31; i++)
    {
        if (nRecDay & nBitFlag)
            strRec = " 1 ";
        else
            strRec = " 0 ";
        OutputDebugString(strRec);
        nBitFlag <<=1;
    }
    OutputDebugString("\r\n");
}
```

See Also

[SearchCalendar](#), [SearchDay](#), [OnSearchDay](#), [ControlID](#)

OnSearchTextCalendar

Description

이 이벤트는 애플리케이션이 SearchTextCalendar()를 호출했을 때 발생하며, 검색 결과를 포함하고 있습니다.

Syntax

```
void OnSearchCalendar(
    long nDeviceID,
    long nControlID,
    long nRecDay
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nRecDay

[in] SearchTextCalendar()에서 지정한 달에 저장된 Text가 있는 날들을 반환합니다. nRecDay의 각 비트는 그 달의 날짜를 나타내며, 가장 우측의 비트(the most-right bit)가 1일입니다. 한 달은 최대 31일까지 있기 때문에, 가장 좌측의 비트(the most-left bit, 32번째 비트)는 0이 되어야 합니다.

예를 들어, nRecDay가 "0001 0000 0000 0000 0000 0000 1011"의 비트 조합을 갖는다면, 1일, 2일, 4일, 29일에 녹화된 영상이 있다는 뜻입니다.

Example

[C++]

```
void CTestSampleDlg::OnSearchTextCalendarXnssdkdevicectrl(long nDeviceID, long nControlID,
long nRecDay)
{
    CString strDay = "01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31\n";
    OutputDebugString(strDay);
```

```

long nBitFlag = 0x01;
CString strRec;

for (int i=0; i<31; i++)
{
    if (nRecDay & nBitFlag)
        strRec = " 1 ";
    else
        strRec = " 0 ";
    OutputDebugString(strRec);
    nBitFlag <<=1;
}
OutputDebugString("Wn");
}

```

See Also

[SearchTextCalendar](#), [SearchText](#), [OnSearchTextList](#), [OnEndSearchText](#), [ControlID](#)

OnSearchOverLap

Description

이 이벤트는 애플리케이션이 SearchOverLap()을 호출했을 때 발생하며, 중복구간 검색 결과를 포함하고 있습니다.

Syntax

```

void OnSearchOverLap(
    long nDeviceID,
    long tStart,
    long tEnd,
    long nTrackNum,
    long nTrackValue
);

```

Parameters

nDeviceID

[in] 장비 ID.

tStart

[in] 검색 시작 시간. (UTC time_t 포맷)

tEnd

[in] 검색 종료 시간. (UTC time_t 포맷)

nTrackNum

[in] 중복구간의 총 개수.

nTrackValue

[in] 각 중복구간의 실제 인식가능한 값

SearchOverLap()에서 지정한 시간의 중복구간 정보를 반환합니다. nTrackNum은 중복구간의 총 개수를 나타내며 nTrackValue는 각 중복구간의 실제 인식가능한 배열 포인터입니다.

예를 들어, nTrackNum이 30이면 nTrackValue 포인터로 nTrackValue[0], nTrackValue[1], nTrackValue[2] 세 개의 실제 중복구간 값을 확인할 수 있습니다.

nTrackNum의 최대 개수는 30개이며, nTrackValue의 최소값은 0입니다.

Example

[C++]

```
void CTestSampleDlg::OnSearchOverLapXnssdkdevicectrl(long nDeviceID, long tStart, long tEnd,
long nTrackNum, long nTrackValue)
{
    unsigned int* trackValue = (unsigned int*)nTrackValue;
    m_nTrackNum = nTrackNum;

    for(int i = 0; i < m_nTrackNum; i++)
    {
        m_nTrackValue[i] = trackValue[i];
    }
    m_nTrackValue[m_nTrackNum] = 'W0';
}
```

See Also

[SearchCalendar](#), [SearchOverLap](#), [SearchDay](#), [OnSearchDay](#)

OnGetStartRecordingTime

Description

이 이벤트는 애플리케이션이 GetStartRecordingTime()을 호출했을 때 발생하며, DVR 녹화 데이터의 시작시간을 포함하고 있습니다.

Syntax

```
void OnGetStartRecordingTime(
    long nDeviceID,
    long tRecUTC
);
```

Parameters

nDeviceID

[in] 장비 ID.

tRecUTC

[in] 녹화 시작 시간. (UTC time_t 포맷)

tEnd

Example

[C++]

```
void CTestSampleDlg::OnGetStartRecordingTimeXnssdkdevicectrl(long nDeviceID, long tRecUTC)
{
    CString str;
    str.Format("OnGetStartRecordingTime.. DeviceID[%d], RecUTC[%d]\n", nDeviceID,
tRecUTC);
    OutputDebugString(str);
}
```

See Also

[GetStartRecordingTime](#), [GetEndRecordingTime](#), [OnGetEndRecordingTime](#)

OnGetEndRecordingTime

Description

이 이벤트는 애플리케이션이 GetEndRecordingTime()을 호출했을 때 발생하며, DVR 녹화 데이터의 종료시간을 포함하고 있습니다.

Syntax

```
void OnGetEndRecordingTime(
    long nDeviceID,
    long tRecUTC
);
```

);

Parameters

nDeviceID

[in] 장비 ID.

tRecUTC

[in] 녹화 종료 시간. (UTC time_t 포맷)

tEnd

Example

[C++]

```
void CTestSampleDlg::OnGetEndRecordingTimeXnssdkdevicectrl(long nDeviceID, long tRecUTC)
{
    CString str;
    str.Format("OnGetEndRecordingTime.. DeviceID[%d], RecUTC[%d]\n", nDeviceID,
tRecUTC);
    OutputDebugString(str);
}
```

See Also

[GetEndRecordingTime](#), [GetStartRecordingTime](#), [OnGetStartRecordingTime](#)

OnSearchDay

Description

이 이벤트는 애플리케이션이 SearchDay()를 호출했을 때 발생하며, 타임라인 핸들이 이벤트 핸들러의 파라미터로 전달됩니다. GetTimeline()과 같은 몇몇 함수들은 이 핸들을 파라미터로 이용합니다.

Syntax

```
void OnSearchDay(
    long nDeviceID,
    long nControlID,
    long hTimeline
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

hTimeline

[in] 타임라인 핸들. 이 값은 Timeline 관련 함수들의 입력 파라미터로 사용됨.

Example

[C++]

```

void CTestSampleDlg::OnSearchDayXnssdkdevicectrl1(long nDeviceID, long nControlID, long
hTimeline)
{
    long CamCount = m_ctlXnsDevice.GetCameraCount(hTimeline);
    long TimelineCount = m_ctlXnsDevice.GetTimelineCount(hTimeline, 0);

    CString strRecInfo;
    long tStart, tEnd;
    long nRecType;
    long nRecId;
    long StartYear, StartMon, StartDay, EndYear, EndMon, EndDay;
    long StartHour, StartMin, StartSec, EndHour, EndMin, EndSec;

    OutputDebugString("<On UTC time>");

    for(int i=0; i<TimelineCount; i++)
    {
        m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
&nRecType, &nRecId);
        long tLocalStart = 0;
        tLocalStart = m_ctlXnsSdkDevice.UTCToDeviceLocalTime(m_hDevice,
tStart);

        m_ctlXnsDevice.TimetToDate(tLocalStart, &StartYear, &StartMon, &StartDay,
&StartHour, &StartMin, &StartSec);

        long tLocalEnd = 0;
        tLocalEnd = m_ctlXnsSdkDevice.UTCToDeviceLocalTime(m_hDevice,
tEnd);

        m_ctlXnsDevice.TimetToDate(tLocalEnd, &EndYear, &EndMon, &EndDay,
&EndHour, &EndMin, &EndSec);
    }
}

```

```

        strRecInfo.Format("index[%d], RecId[%d], RecType[%d], time[%04d-%02d-
%02d %02d:%02d:%02d ~ %04d-%02d-%02d %02d:%02d:%02d]₩n", i, nRecId, nRecType,
StartYear, StartMon, StartDay, StartHour, StartMin, StartSec, EndYear, EndMon, EndDay, EndHour,
EndMin, EndSec);

        OutputDebugString(strRecInfo);
    }
}

```

See Also

[SearchDay](#), [OnSearchCalendar](#), [SearchCalendar](#), [ControlID](#), [UTC Time](#), [LocalTimeToDeviceUTC](#)

OnGetMotionGrid

Description

이 이벤트는 애플리케이션이 GetMotionGrid()을 호출했을 때 발생하며, DVR 장비의 Grid 정보를 포함하고 있습니다.

Syntax

```

void OnGetMotionGrid(
    long nDeviceID,
    long nControlID,
    long nGridWidth,
    long nGridHeight,
    long pGridArray
);

```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nGridHeight

[in] 가로 Grid 개수(최대 64).

nGridWidth

[in] 세로 Grid 개수(최대 64).

pGridArray

[in] nGridWidth X nGridHeight 개수만큼의 char 배열값(최대 64 X 64). 설정 가능한 값만큼 '1'로 마킹되어 있음.

Ex) 3 X 3 의 pGridArray일경우 화면영역의 Grid는 아래와 같음

[0][0]	[0][1]	[0][2]
[1][0]	[1][1]	[1][2]
[2][0]	[2][1]	[2][2]

Example

[C++]

```
void CTestSampleDlg::OnGetMotionGridXnssdkdevicectrl(long nDeviceID, long nControlID, long
nGridWidth, long nGridHeight, long pGridArray)
{
    CString str;
    char m_pGridArray[64][64];
    m_nGridWidth = nGridWidth;
    m_nGridHeight = nGridHeight;

    memcpy(m_pGridArray, (char*)pGridArray, 64*64);
    for(int i = 0; i < nGridHeight; i++)
    {
        m_pGridArray[i][m_nGridWidth] = 'W0';
    }
    m_pGridArray[m_nGridHeight][0] = 'W0';

    str.Format("Event OnGetMotionGrid.. nGridWidth[%d], nGridHeight[%d]Wn", nGridWidth,
nGridHeight);
    OutputDebugString(str);
}
```

See Also

[GetMotionGrid](#), [MotionSearch](#), [OnMotionSearch](#)

OnMotionSearch

Description

이 이벤트는 애플리케이션이 MotionSearch()를 호출했을 때 발생하며, 타임라인 핸들이 이벤트 핸들러의 파라미터로 전달됩니다. GetTimeline()과 같은 몇몇 함수들은 이 핸들을 파라미터로 이용합니다.

Syntax

```
void OnMotionSearch(
    long nDeviceID,
    long nControlID,
    long hTimeline
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

hTimeline

[in] 타임라인 핸들. 이 값은 Timeline 관련 함수들의 입력 파라미터로 사용됨.

Example

[C++]

```
void CTestSampleDlg::OnMotionSearchXnssdkdevicectrl(long nDeviceID, long nControlID, long
hTimeline)
{
    long CamCount = m_ctlXnsDevice.GetCameraCount(hTimeline);
    long TimelineCount = m_ctlXnsDevice.GetTimelineCount(hTimeline, 0);

    CString strRecInfo;
    long tStart, tEnd;
    long nRecType;
    long nRecId;
    long StartYear, StartMon, StartDay, EndYear, EndMon, EndDay;
    long StartHour, StartMin, StartSec, EndHour, EndMin, EndSec;
```

```

        OutputDebugString("<On UTC time>");

        for(int i=0; i<TimelineCount; i++)
        {
            m_ctlXnsDevice.GetTimeline(hTimeline, 0, i, XTIME_UTC, &tStart, &tEnd,
            &nRecType, &nRecId);

            long tLocalStart = 0;
            tLocalStart = m_ctrlXnsSdkDevice.UTCToDeviceLocalTime(m_hDevice, tStart);
            m_ctlXnsDevice.TimetToDate(tLocalStart, &StartYear, &StartMon, &StartDay,
            &StartHour, &StartMin, &StartSec);

            long tLocalEnd = 0;
            tLocalEnd = m_ctrlXnsSdkDevice.UTCToDeviceLocalTime(m_hDevice, tEnd);
            m_ctlXnsDevice.TimetToDate(tLocalEnd, &EndYear, &EndMon, &EndDay,
            &EndHour, &EndMin, &EndSec);

            strRecInfo.Format("index[%d], RecId[%d], RecType[%d], time[%04d-%02d-
            %02d %02d:%02d:%02d ~ %04d-%02d-%02d %02d:%02d:%02d]₩n", i, nRecId, nRecType,
            StartYear, StartMon, StartDay, StartHour, StartMin, StartSec, EndYear, EndMon, EndDay, EndHour,
            EndMin, EndSec);
            OutputDebugString(strRecInfo);
        }
    }
}

```

See Also

[MotionSearch](#), [GetMotionGrid](#), [OnGetMotionGrid](#), [OnSearchCalendar](#), [SearchCalendar](#), [ControlID](#), [UTC Time](#), [LocalTimeToDeviceUTC](#)

OnSearchTextList

Description

이 이벤트는 SearchText()를 호출했을 때 발생하며, 장비로부터 text 검색 결과를 받는동안 주기적으로 발생합니다.

Syntax

```

void OnSearchTextList(
    long nDeviceID,
    long nControlID,
    BSTR szLinkChannel,
)

```

```

        long tEventTime,
        long tPlayTime,
        long nTotalCount,
        BSTR szText
);

```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

szLinkchannel

[in] Control module의 카메라 link 상태.

tEventTime

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

tPlayTime

[in] 이벤트 발생에 대해 재생해야 할 시간. (4바이트의 time_t 포맷임.)

nTotalCount

[in] 총 text 개수

szText

[in] 검색된 text

Example

[C++]

```

void CTestSampleDlg::OnSearchTextListXnssdkdevicectrl(long nDeviceID, long nControlID, BSTR
szLinkChannel, long tEventTime, long tPlayTime, long nTotalCount, BSTR szText)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventTime, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("OnSearchTextList Event DeviceID[%d] ControlID[%d], EventTime[%d-%d-
%d %02d:%02d:%02d] TotalCount[%d] Text[%s]₩n", nDeviceID, nControlID, Year, Mon, Day, Hour,
    Min, Sec, nTotalCount, szText);
    OutputDebugString(str);
}

```

See Also

[SearchText](#), [OnEndSearchText](#), [SearchTextCalendar](#), [ControlID](#), [UTC Time](#), [LocalTimeToDeviceUTC](#)

OnEndSearchText

Description

이 이벤트는 장비로부터 text 검색 결과를 전부 받았을 때 발생합니다. 에러 코드를 포함합니다.

Syntax

```
void OnEndSearchText(
    long nDeviceID,
    long nControlID,
    long nErrorCode,
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nErrorCode

[in] 에러 코드

Example

[C++]

```
void CTestSampleDlg::OnEndSearchTextXnssdkdevicectrl(long nDeviceID, long nControlID, long
nErrorCode)
{
    CString str;
    str.Format("OnEndSearchText Event DeviceID[%d] ControlID[%d] Error[%s]\n", nDeviceID,
nControlID, m_ctlXnsDevice.GetErrorString(nErrorCode));
    OutputDebugString(str);
}
```

See Also

[SearchText](#), [OnSearchTextList](#), [SearchTextCalendar](#), [ControlID](#), [UTC Time](#), [LocalTimeToDeviceUTC](#)

OnGetPresetList

Description

이 이벤트는 GetPresetList()가 호출되면 발생하며, 프리셋 리스트의 핸들은 이벤트 핸들러의 파라미터로 전달됩니다. 이 핸들은 GetPresetCount()의 파라미터로 입력되며, 프리셋과 관련된 정보를 얻는데 사용됩니다.

Syntax

```
void OnGetPresetList(
    long nDeviceID,
    long nControlID,
    long hPresetListList
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

hPresetListList

[in] 프리셋 리스트를 가리키는 핸들.

Example

[C++]

```
void CTestSampleDlg::OnGetPresetListXnssdkdevicectrl1(long nDeviceID, long nControlID, long
hPresetListList)
{
    CString str;
    long nNumber;
    long nPresetCount;
    CString strPresetName;

    nPresetCount = m_ctlXnsDevice.GetPresetCount(hPresetListList);
    str.Format("Event OnGetPresetList.. nDevice[%d], nControlId[%d], PresetCount[%d]\n",
nDeviceID, nControlID, nPresetCount);
```

```

        OutputDebugString(str);

        for(int i=0; i<nPresetCount; i++)
        {
            strPresetName = m_ctlXnsDevice.GetPreset(hPresetListList, i, &nNumber);
            str.Format("PresetName[%s], nPresetNumber[%d]\n", strPresetName,
nNumber);
            OutputDebugString(str);
        }
    }
}

```

See Also

[GetPresetList](#), [GetPresetCount](#), [ControlID](#)

OnUploadSoftwareProgress

Description

이 이벤트는 UploadSoftware()가 호출되어 소프트웨어가 장비로 업로드 되는 동안 진척도가 5% 증가할 때마다 주기적으로 발생합니다. 진척도(%)와 에러 코드를 포함합니다.

Syntax

```

void OnUploadSoftwareProgress(
    long nDeviceID,
    long nErrorCode,
    long nProgress
);

```

Parameters

nDeviceID

[in] 장비 ID.

nErrorCode

[in] 에러 코드.

nProgress

[in] 진척도 (단위: %)

Example

[C++]

```
void CTestSampleDlg::OnUploadSoftwareProgressXnssdkdevicectrl1(long nDeviceID, long
nErrorCode, long nProgress)
{
    char str[200];
    sprintf(str, "Upload Software DeviceID[%d] Progress[%d%], ErrCode[%s]\n", nDeviceID,
nProgress, m_ctlXnsDevice.GetErrorString(nErrorCode));
    OutputDebugString(str);
}
```

See Also

[UploadSoftware](#)

OnImportConfigProgress

Description

이 이벤트는 ImportConfig()를 이용하여 장비로부터 설정 파일을 가져올 때 발생합니다. 진척도가 5% 증가할 때마다 주기적으로 발생하며 에러 코드를 포함합니다.

Syntax

```
void OnImportConfigProgress(
    long nDeviceID,
    long nErrorCode,
    long nProgress
);
```

Parameters

nDeviceID

[in] 장비 ID.

nErrorCode

[in] 에러 코드.

nProgress

[in] 진척도 (단위: %)

Example

[C++]

```
void CTestSampleDlg::OnImportConfigProgressXnssdkdevicectrl1(long nDeviceID, long
nErrorCode, long nProgress)
```

```
{
    char str[200];
    sprintf(str, "Import Config DeviceID[%d] Progress[%d%], ErrCode[%s]\n", nDeviceID,
nProgress, m_ctlXnsDevice.GetErrorString(nErrorCode));
    OutputDebugString(str);
}
```

See Also

[ImportConfig](#)

OnExportConfigProgress

Description

이 이벤트는 ExportConfig()를 이용하여 장비로 설정 파일을 내보낼 때 진척도가 5% 증가할 때마다 주기적으로 발생합니다. 진척도(%)와 에러 코드를 포함합니다.

Syntax

```
void OnExportConfigProgress(
    long nDeviceID,
    long nErrorCode,
    long nProgress
);
```

Parameters

nDeviceID

[in] 장비 ID.

nErrorCode

[in] 에러 코드.

nProgress

[in] 진척도 (단위: %)

Example

[\[C++\]](#)

```
void CTestSampleDlg::OnExportConfigProgressXnssdkdevicectrl1(long nDeviceID, long
nErrorCode, long nProgress)
{
    char str[200];
```

```

        sprintf(str, "Export Config DeviceID[%d] Progress[%d%], ErrCode[%s]\n", nDeviceID,
nProgress, m_ctlXnsDevice.GetErrorString(nErrorCode));
        OutputDebugString(str);
}

```

See Also

[ExportConfig](#)

OnDeviceStatusChanged

Description

장비의 상태가 변경되었을 때, 발생하는 이벤트입니다. Connect()를 이용하여 장비에 접속하거나 접속이 해제되어 재접속할 때 발생합니다. 재접속 시에는 세 번째 인자인 nDeviceStatus 값이 true입니다. 이 경우 OpenMedia() 또는 OpenStream()을 호출하여 미디어를 다시 open 해주어야 합니다.

Syntax

```

void OnDeviceStatusChanged(
    long nDeviceID,
    long nErrorCode,
    long nDeviceStatus,
    long nHddCondition
);

```

Parameters

nDeviceID

[in] 장비 ID.

nErrorCode

[in] 에러 코드.

nDeviceStatus

[in] 장비의 상태.

nHddCondition

[in] (사용하지 않음.)

Example

[C++]

```
void CTestSampleDlg::OnDeviceStatusChangedXnssdkdevicectrl1(long nDeviceID, long
```

```
nErrorCode, long nDeviceStatus, long nHddCondition)
{
    char str[200];
    if (nErrorCode == ERR_SUCCESS && nDeviceStatus == 1)
    {
        if (m_ctlXnsDevice.GetDeviceStatus(m_hDevice) == XDEVICE_STATUS_CONNECTED )
            m_ctlXnsDevice.OpenMedia(m_hDevice, 2, XMEDIA_LIVE, 0, 0,
                                     &m_pMediaLiveSource);
    }
    sprintf(str, "OnDevice Status Changed DeviceID[%d], nDeviceStatus[%d%],
nHddCondition[%d], ErrCode[%s]#\n", nDeviceID, nDeviceStatus, nHddCondition,
m_ctlXnsDevice.GetErrorString(nErrorCode));

    OutputDebugString(str);
}
```

See Also

[Connect](#)

OnControlStatusChanged

Description

이 이벤트는 장비의 연결 상태나 control module의 기능(capability)이 변경되었을 때 발생합니다. 애플리케이션은 이 이벤트를 받았을 때, GetControlStatus()를 이용하여 변경된 내용을 확인 할 수 있습니다.

Syntax

```
void OnControlStatusChanged(
    long nDeviceID,
    long nControlID,
    long nErrorCode,
    long nStatusMask
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nErrorCode

[in] 에러 코드.

nStatusMask

[in] Control module의 어떤 상태 정보가 변경되었는지 나타냄. (표 2.12 참조)

표 2.12 Control Module Status

Status	Value	Description
XCAMERA_STATUS_MASK	0x00000001	카메라 상태.
XRECORDING_STATUS_MASK	0x00000002	녹화 상태.
XLISTEN_STATUS_MASK	0x00000004	Listen 기능의 동작 상태.
XTALK_STATUS_MASK	0x00000008	Talk 기능의 동작 상태.
XALARM_STATUS_MASK	0x00000010	알람 상태.
XLOCALREC_STATUS_MASK	0x00000020	로컬 PC에서의 녹화 상태.

Example

[C++]

```
void CTestSampleDlg::OnControlStatusChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nErrorCode, long nStatusMask)
{
    char str[200];
    sprintf(str, "Control Status Changed DeviceID[%d] ControlID[%d], Status[%d],
    ErrorCode[%s]\n", nDeviceID, nControlID, nStatusMask,
    m_ctlXnsDevice.GetErrorString(nErrorCode));

    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnControlCapabilityChanged

Description

Control module의 기능(capability)이 변경되었을 때 발생하는 이벤트입니다.

Syntax

```
void OnControlCapabilityChanged(
    long nDeviceID,
    long nControlID,
    long nCapabilityMask
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nCapabilityMask

[in] Control module의 어떤 기능이 변경되었는지 나타냄. (표 2.13 참조)

표 2.13 Control Module Capability

Field	Value	Description
XCTL_CAP_SW_UPLOAD_MASK	0x0000000001	원격으로 펌웨어를 업데이트 함.
XCTL_CAP_CFG_IMPORT_MASK	0x0000000002	장비로부터 설정 파일을 가져옴.
XCTL_CAP_CFG_EXPORT_MASK	0x0000000004	장비로 설정 파일을 보냄.
XCTL_CAP_CONN_FORCED_MASK	0x0000000008	관리자(admin) 계정으로 강제 접속함.
XCTL_CAP_MEDIA_AUTHORITY_MASK	0x0000000010	미디어의 재생 권한을 얻음.
XCTL_CAP_PLAY_CONTROL_MASK	0x0000000020	재생을 제어하는 기능.
XCTL_CAP_LIVE_MASK	0x0000000040	실시간 영상을 재생.
XCTL_CAP_PLAYBACK_MASK	0x0000000080	영상 재생.
XCTL_CAP_BACKUP_MASK	0x0000000100	영상 백업 기능.
XCTL_CAP_SEARCH_MASK	0x000000200	영상 검색 기능.
XCTL_CAP_LISTEN_MASK	0x000000400	Listen 기능.
XCTL_CAP_TALK_MASK	0x000000800	Talk 기능.
XCTL_CAP_PTZ_MASK	0x000001000	PTZ 제어 기능.
XCTL_CAP_ALARM_ONOFF_MASK	0x000002000	Alarm-Out On/Off 기능.
XCTL_CAP_DVR_EVENT_RESET_MASK	0x000004000	DVR 의 이벤트 표시를 없애는 기능
XCTL_CAP_DVR_MANUAL_REC CONTR	0x000008000	원격지에서 DVR 내부의 녹화를

OL_MASK		제어하는 기능
XCTL_CAP_DVR_REC_STATUS_MASK	0x00010000	DVR 녹화 상태를 확인하는 기능.

Example

[C++]

```
void CTestSampleDlg::OnControlCapabilityChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nCapabilityMask)
{
    char str[200];
    sprintf(str, "Control Capability Changed DeviceID[%d] ControlID[%d], Status[%d]\n",
            nDeviceID, nControlID, nCapabilityMask);
    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnMotionEvent

Description

이 이벤트는 장비에서 모션 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnMotionEvent(
    long nDeviceID,
    long nControlID,
    long tEventUTC
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CTestSampleDlg::OnMotionEventXnssdkdevicectrl1(long nDeviceID, long nControlID, long tEventUTC)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("Motion Event DeviceID[%d] ControlID[%d], time[%d-%d-%d %02d:%02d:%02d]\r\n", nDeviceID, nControlID, Year, Mon, Day, Hour, Min, Sec);
    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnSensorEvent

Description

이 이벤트는 장비에서 sensor-in(즉, alarm-in) 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnSensorEvent(
    long nDeviceID,
    long nControlID,
    long nSensorNumber,
    long tEventUTC
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nSensorNumber

[in] 이벤트가 발생한 센서의 번호.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CTestSampleDlg::OnSensorEventXnssdkdevicectrl1(long nDeviceID, long nControlID, long
nSensorNumber, long tEventUTC)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("SensorEvent DeviceID[%d] ControlID[%d], SensorInput[%d], time[%d-%d-
%d %02d:%02d:%02d]\n", nDeviceID, nControlID, nSensorNumber, Year, Mon, Day, Hour, Min,
Sec);
    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnIvOccurEvent

Description

이 이벤트는 장비에서 IV 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnIvOccurEvent (
    long nDeviceID,
    long nControlID,
    long tEventUTC,
    long nIvType
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CIV_TESTDlg::OnIvOccurEventXnssdkdevicectrl1(long nDeviceID, long nControlID, long
tEventUTC, long nIvType)
{
    // TODO: Add your message handler code here
    long year, mon, day, hour, min, sec;

    m_ctlXnsDevice.TimetToDate(tEventUTC, &year, &mon, &day, &hour, &min, &sec);
    DBG_LOG(_T("$$IV Event Detected, device=%d, cid=%d, time=[%04d-%02d-
%02d %02d:%02d:%02d]\n"),
            nDeviceID, nControlID, year, mon, day, hour, min, sec);
    if(nIvType & XIV_TYPE_PASSING)
        DBG_LOG(_T("$$IV TYPE : PASSING\n"));
    if(nIvType & XIV_TYPE_ENTERING)
        DBG_LOG(_T("$$IV TYPE : ENTERING\n"));
    if(nIvType & XIV_TYPE_EXITING)
        DBG_LOG(_T("$$IV TYPE : EXITING\n"));
    if(nIvType & XIV_TYPE_APPEARING)
        DBG_LOG(_T("$$IV TYPE : APPEARING\n"));
    if(nIvType & XIV_TYPE_DISAPPEARING)
        DBG_LOG(_T("$$IV TYPE : DISAPPEARING\n"));
    if(nIvType & XIV_TYPE_SCENE_CHANGE)
        DBG_LOG(_T("$$IV TYPE : SCENE_CHANGE\n"));
}
```

See Also

[ControlID](#)

OnVideoLossEvent

Description

이 이벤트는 장비에서 영상 손실(video loss) 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnVideoLossEvent(
    long nDeviceID,
    long nControlID,
    long tEventUTC
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CTestSampleDlg::OnVideoLossEventXnssdkdevicectrl1(long nDeviceID, long nControlID,
long tEventUTC)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("VideoLossEvent DeviceID[%d] ControlID[%d], time[%d-%d-
%d %02d:%02d:%02d]\n", nDeviceID, nControlID, Year, Mon, Day, Hour, Min, Sec);
    OutputDebugString(str);
}
```

See Also

[ControlID](#), [OnVideoLossEnd](#)

OnVideoLossEnd

Description

이 이벤트는 장비에서 영상 손실(video loss) 이벤트가 종료했음을 알려줍니다.

해당 채널의 영상을 보고 있을 때 지원하며, 영상을 보고 있지 않을 경우에는 일부 장비에서만

지원됩니다. (DVR)

Syntax

```
void OnVideoLossEnd(
    long nDeviceID,
    long nControlID,
    long tEventUTC
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CTestSampleDlg::OnVideoLossEndXnssdkdevicectrl1(long nDeviceID, long nControlID, long
tEventUTC)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("VideoLossEndEvent DeviceID[%d] ControlID[%d], time[%d-%d-
%d %02d:%02d:%02d]\n", nDeviceID, nControlID, Year, Mon, Day, Hour, Min, Sec);
    OutputDebugString(str);
}
```

See Also

[ControlID](#), [OnVideoLossEvent](#)

OnTextOccur

Description

이 이벤트는 장비에서 text 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnTextOccur(
    long nDeviceID,
    long nControlID,
    long tEventUTC,
    long tPlayUTC,
    BSTR szValue
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

tPlayUTC

[in] 이벤트 발생에 대해 재생해야 할 시간. (4바이트의 time_t 포맷임.)

szValue

[in] text value

Example

[C++]

```
void CTestSampleDlg::OnTextOccurXnssdkdevicectrl(long nDeviceID, long nControlID, long
tEventUTC, long tPlayUTC, BSTR szValue)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("OnTextOccur Event DeviceID[%d] ControlID[%d], EventTime[%d-%d-
%d %02d:%02d:%02d] %w\ n", nDeviceID, nControlID, Year, Mon, Day, Hour, Min, Sec);
    OutputDebugString(str);
```

```
}
```

See Also

[ControlID](#)

OnHDDFailed

Description

이 이벤트는 DVR내의 하드디스크의 기록 실패, 인식 실패, 재생 실패 등 문제가 생겼을 때 발생합니다.

Syntax

```
void OnHDDFailed(
    long nDeviceID,
    long nHDDNumber
);
```

Parameters

nDeviceID

[in] 장비 ID.

nHDDNumber

[in] 하드디스크 번호.

Example

[C++]

```
void CTestSampleDlg::OnHDDFailedXnssdkdevicectrl1(long nDeviceID, long nHDDNumber)
{
    CString str;
    str.Format("HDD Failed DeviceID[%d], nHDDNumber[%d]\n", nDeviceID, nHDDNumber);
    OutputDebugString(str);
}
```

OnHDDFulled

Description

이 이벤트는 DVR 내의 하드디스크에 용량이 다 찼을 때 발생합니다. DVR에 덮어쓰기(overwrite) 옵션이 활성화되어 있다면, 이 이벤트는 발생하지 않습니다.

Syntax

```
void OnHDDFulled(
    long nDeviceID,
    long nHDDNumber
);
```

Parameters

nDeviceID

[in] 장비 ID.

nHDDNumber

[in] 하드디스크 번호.

Example

[C++]

```
void CTestSampleDlg::OnHDDFulledXnssdkdevicectrl1(long nDeviceID, long nHDDNumber)
{
    CString str;
    str.Format("HDD Fulled DeviceID[%d], nHDDNumber[%d]\r\n", nDeviceID, nHDDNumber);
    OutputDebugString(str);
}
```

OnHDDNumChanged

Description

이 이벤트는 DVR 내의 하드디스크 개수가 줄어들었을 때 발생합니다. 하드디스크 개수가 증가했을 때는 발생하지 않습니다.

Syntax

```
void OnHDDNumChanged(
    long nDeviceID,
);
```

Parameters

nDeviceID

[in] 장비 ID.

Example

[C++]

```
void CTestSampleDlg::OnHDDNumChangedXnssdkdevicectrl(long nDeviceID)
{
    CString str;
    str.Format("HDD NumChanged DeviceID[%d]\n", nDeviceID);
    OutputDebugString(str);
}
```

OnPasswordChanged

Description

이 이벤트는 DVR의 비밀번호가 변경되었을 때 발생합니다.

Syntax

```
void OnPasswordChanged(
    long nDeviceID,
    BSTR szID
);
```

Parameters

nDeviceID

[in] 장비 ID.

szID

[in] 비밀번호가 변경된 ID.

Example

[C++]

```
void CTestSampleDlg::OnPasswordChangedXnssdkdevicectrl1(long nDeviceID, LPCTSTR szID)
{
    char str[200];
    sprintf(str, "PasswordChanged DeviceID[%d] szID[%s]\n", nDeviceID, szID);
    OutputDebugString(str);
}
```

OnBackupProgress

Description

이 이벤트는 장비로부터 미디어 데이터를 백업하는 동안 진척도가 5% 증가할 때마다 주기적으로 발생합니다. 진척도(%)와 에러 코드를 포함합니다.

Syntax

```
void OnBackupProgress(
    long nDeviceID,
    long nControlID,
    long nErrorCode,
    long nProgress,
    long tStart,
    long tEnd
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nErrorCode

[in] 에러 코드.

nProgress

[in] 진척도. (단위: %)

tStart

[in] 백업 시작 시간. (4바이트의 time_t 포맷임.)

tEnd

[in] 백업 종료 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CTestSampleDlg::OnBackupProgressXnssdkdevicectrl1(long nDeviceID, long nControlID,
long nErrorCode, long nProgress, long tStart, long tEnd)
{
```

```

CString str;
long StartYear, StartMon, StartDay, StartHour, StartMin, StartSec;
long EndYear, EndMon, EndDay, EndHour, EndMin, EndSec;

m_ctlXnsDevice.TimetToDate(tStart, &StartYear, &StartMon, &StartDay, &StartHour,
&StartMin, &StartSec);

m_ctlXnsDevice.TimetToDate(tEnd, &EndYear, &EndMon, &EndDay, &EndHour, &EndMin,
&EndSec);

str.Format("Backup Progress DeviceID[%d] ControlID[%d], nProgress[%d], StartTime[%d-
%d-%d %02d:%02d:%02d], EndTime[%d-%d-%d %02d:%02d:%02d], ErrorCode[%s]\n",
nDeviceID, nControlID, nProgress, StartYear, StartMon, StartDay, StartHour, StartMin, StartSec,
EndYear, EndMon, EndDay, EndHour, EndMin, EndSec,
m_ctlXnsDevice.GetErrorString(nErrorCode));

OutputDebugString(str);
}

```

See Also

[StartBackup](#), [ControlID](#)

OnNewMedia

Description

프레임 데이터가 수신되면 발생합니다. 애플리케이션이 OpenStream() 또는 OpenText()를 이용하여 스트림 데이터를 요청하면, 스트림 데이터는 이 이벤트를 통해 전달됩니다.

Syntax

```

void OnNewMedia(
    long nDeviceID,
    long nControlID,
    long nMediaType,
    long hMediaData,
    long nInputType,
    long nFrameType,
    long nFrameTime,
    long nFrameSize,
    long nTimeStamp
);

```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nMediaType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live
- XMEDIA_PLAYBACK (=2): Playback
- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)
- XMEDIA_TEXT_LIVE (=5): Live(text)
- XMEDIA_TEXT_PLAYBACK (=6): Playback(text)
- XMEDIA_TEXT_BACKUP (=7): Backup replay (local PC에 Text 데이터를 저장함)

hMediaData

[in] 프레임 데이터의 핸들. 이 데이터를 화면에 출력하려면 XnsSdkWindow::PushMedia()를 호출하고 이 핸들을 파라미터로 입력함.

nInputType

[in] 입력 타입.

- XINPUT_VIDEO (=1): 비디오 데이터.
- XINPUT_AUDIO (=2): 오디오 데이터
- XINPUT_METADATA (=3): 메타 데이터

nFrameType

[in] 프레임 타입. 다음의 타입은 *nInputType*가 XINPUT_VIDEO인 경우에만 유효함.

- XFRAME_UNKNOWN (=0)
- XFRAME_IVOP (=1): Intra Video Object Plane
- XFRAME_PVOP (=2): Predictive VOP
- XFRAME_BVOP (=3): Bidirectional VOP
- XFRAME_END (=4)

nFrameTime

[in] 프레임의 생성 시간.

nFrameSize

[in] 프레임 길이.

nTimeStamp

[in] 프레임 재생 시간. (단위: millisecond) 현재 프레임이 이전 프레임으로부터 몇 ms 뒤의 영상인지 확인하는데 사용. 예를 들어, 첫 번째 프레임이 3042이고 두 번째 프레임이 30ms 이후의 영상이라면 두 번째 프레임의 nTimeStamp는 3072 값을 가짐.

Remarks

hMediaData는 미디어 스트림 버퍼에 대한 첫번째 주소를 가르킵니다. (C++의 포인터와 동일합니다.) hMediaData의 처음 296byte는 헤더정보이며, 인코딩 데이터는 그 다음에 옵니다.

Example

[C++]

```
void CTestSampleDlg::OnNewMediaXnssdkdevicectrl1(long nDeviceID, long nControlID, long nMediaType, long hMediaData, long nInputType, long nFrameType, long nFrameTime, long nFrameSize, long nTimeStamp)
{
    // this function draw video on the window control
    m_ctlXnsWindow.PushMedia(hMediaData);

    // or you can use the encoded video data as follows
    char *video_header = (char *) hMediaData;
    char *video_data = video_header+296;

    /* video header information
    typedef struct xns_video_header2_tag
    {
        char vendor_name[8];      //SVSSVH, SVSSMH, SVSSAH
        unsigned long major_ver;
        unsigned long minor_ver;
        unsigned long header_size;
        unsigned long media_type; //XNSE_MEDIA_TYPE
        unsigned long camera_index;
        unsigned long codec_type; //XNSE_MEDIA_CODEC_TYPE
        unsigned long replay_sequence;
        unsigned long time_stamp;
        unsigned long frame_size;
        unsigned long frame_type; //MPEG_FRAME_TYPE
        unsigned char igrp_group_count;
        unsigned char pgp_group_count;
        unsigned char exist_bframe;
        unsigned char bSkip;
        unsigned long frame_rate;
    }
```

```

        unsigned long      is_audiosrc;
        unsigned long      is_videoloss;
        unsigned long      is_alarm;
        unsigned long      is_motion;
        unsigned long      is_motion_region;
        unsigned long      frame_width;
        unsigned long      frame_height;
        TIME_ZONE_INFORMATION time_zone;
        long               rec_time;
        unsigned long      water_mark[4];
        unsigned short     is_ivevent
        unsigned short     reserveds;
        unsigned long      reserved[2];
        unsigned long      record_id;
        unsigned long      water_mark_type;
    } XNS_VIDEO_HEADER2;
*/
/* audio header information
typedef struct xns_audio_header2_tag
{
    char              vendor_name[8];
    unsigned long     major_ver;
    unsigned long     minor_ver;
    unsigned long     header_size;
    unsigned long     media_type;
    unsigned long     mic_index;
    unsigned long     codec_type;
    unsigned long     time_stamp;
    unsigned long     frame_size;
    TIME_ZONE_INFORMATION   time_zone;
    unsigned long     channel      : 2;
    unsigned long     sampling     : 6;
    unsigned long     bits         : 4;      //AAC Codec
    unsigned long     m_cConsecutiveSeq : 8;  //RTP header Extension (CSeq)
    unsigned long     reserved     : 12;
} XNS_AUDIO_HEADER2;
*/
/* metadata header information
typedef struct xns_metadata_header_tag

```

```
{
    char          vendor_name[8];      //SVSSVH, SVSSMH, SVSSAH
    unsigned long major_ver;
    unsigned long minor_ver;
    unsigned long header_size;
    unsigned long media_type;        //XNSE_MEDIA_TYPE //MEDIA_TYPE2
    unsigned long time_stamp;
    unsigned long camera_index;
    unsigned long frame_size;
    TIME_ZONE_INFORMATION time_zone;
    unsigned long metadata_type;
    unsigned char   frame_type;
    unsigned char   reserved1[3];
    unsigned long   reserved2[2];
} XNS_METADATA_HEADER;
*/
```

XNSE_MEDIA_TYPE

```
typedef enum
{
    XNSE_MTYPE_NONE          = 0X00,
    XNSE_MTYPE_LIVE           = 0x01,
    XNSE_MTYPE_SEARCH_NET     = 0x02,
    XNSE_MTYPE_REPLAY          = 0x04,
    XNSE_MTYPE_SEARCH_LOCAL   = 0x08,
    XNSE_MTYPE_SEARCH_CLIP    = 0x10,
    XNSE_MTYPE_BACKUP_NET      = 0x20,
    XNSE_MTYPE_BACKUP_LOCAL    = 0x40,
    XNSE_MTYPE_COPY             = 0x80,

    XNSE_MTYPE_METADATA         = 0x0100,
    XNSE_MTYPE_METADATA_SEARCH  = 0x0200,
    XNSE_MTYPE_METADATA_BACKUP  = 0x0400,
} XNSE_MEDIA_TYPE;
```

XNSE_MEDIA_CODEC_TYPE

```
typedef enum
```

```
{
    XNSE_CODEC_UNKNOWN      = 0x00000000,
    XNSE_CODEC_JPEG         = 0x00000001,
    XNSE_CODEC_MJPEG        = 0x00000002,
    XNSE_CODEC_MP4V         = 0x00000004,
    XNSE_CODEC_MP4V_ASP     = 0x00000008,
    XNSE_CODEC_PENTA        = 0x00000010,
    XNSE_CODEC_SOLO         = 0x00000020,
    XNSE_CODEC_H264          = 0x00000040,
    XNSE_CODEC_JPEG2000     = 0x00000080,
    XNSE_CODEC_IDIS_MPEG4    = 0x00000100,
    XNSE_CODEC_IDIS_H264     = 0x00000200,
    XNSE_CODEC_G711          = 0x00010000,
    XNSE_CODEC_G726          = 0x00020000,
    XNSE_CODEC_PCM           = 0x00040000,
    XNSE_CODEC_G726_504X     = 0x00080000,
    XNSE_CODEC_IMA_ADPCM     = 0x00100000,
    XNSE_CODEC_G723          = 0x00200000,
    XNSE_CODEC_G723_SVR      = 0x00400000,
    XNSE_CODEC_PCM_WESP      = 0x00800000,
    XNSE_CODEC_G726_SNR      = 0x01000000,
    XNSE_CODEC_DVI4_HISILICON = 0x02000000,
    XNSE_CODEC_G711U_HISILICON = 0x04000000,
    XNSE_CODEC_G726_VER       = 0x08000000,
    XNSE_CODEC_G711_IPCAM     = 0x10000000,
    XNSE_CODEC_AAC           = 0x20000000,
} XNSE_MEDIA_CODEC_TYPE;
```

XNSE_MEDIA_TYPE

```
typedef enum
{
    XNSE_MPEG_UNKNOWN = 0,
    XNSE_MPEG_IVOP,
    XNSE_MPEG_PVOP,
    XNSE_MPEG_BVOP,
    XNSE_BACKUP_END,
    XNSE_THUMBNAIL,
    XNSE_THUMBNAIL_END,
```

```
XNSE_BACKUP_FAILED,
XNSE_BACKUP_CHUNK_END,
} MPEG_FRAME_TYPE;           // frame type
```

See Also

[OpenStream](#), [XnsSdkWindow::PushMedia](#), [ControlID](#)

OnFanBroken

Description

카메라의 팬(fan)이 고장 나면 발생하는 이벤트입니다.

Syntax

```
void OnFanBroken(long nDeviceID);
```

Parameters

nDeviceID

[in] 장비 ID.

Example

[C++]

```
void CTestSampleDlg::OnFanBrokenXnssdkdevicectl1(long nDeviceID)
{
    CString str;
    str.Format("Fan Broken DeviceID[%d]\n", nDeviceID);
    OutputDebugString(str);
}
```

OnFanStatusChanged

Description

DVR/NVR 팬(fan)의 상태변경이 생기면 발생하는 이벤트입니다.

Syntax

```
void OnFanStatusChanged(long nDeviceID, long tEventUTC, long nFanNo, long nStatus);
```

Parameters

nDeviceID

[in] 장비 ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

nFanNo

[in] 상태가 변경된 Fan의 번호.

nStatus

[in] Fan의 변경된 상태를 나타냄.

- (1): 고장으로 변경.

- (0): 정상으로 변경.

Example

[C++]

```
void CTestSampleDlg::OnFanStatusChangedXnssdkdevicectrl1(long nDeviceID, long tEventUTC,
long nFanNo, long nStatus)
{
    CString str;
    str.Format("Fan Status Changed DeviceID[%d], FanNo[%d], Status[%d] \n", nDeviceID,
nFanNo, nStatus);
    OutputDebugString(str);
}
```

OnPtzControlFailed

Description

PTZ 제어에 실패했을 때 발생하는 이벤트입니다.

Syntax

```
void OnPtzControlFailed(
    long nDeviceID,
    long nControlID,
    long nErrorCode
);
```

Parameters

nDeviceID
 [in] 장비 ID.
nControlID
 [in] Control ID.
nErrorCode
 [in] 에러 코드.

Example

[C++]

```
void CTestSampleDlg::OnPtzControlFailedXnssdkdevicectrl1(long nDeviceID, long nControlID,
long nErrorCode)
{
    char str[200];
    sprintf(str, "Ptz Control Failed DeviceID[%d] ControlID[%d], nErrorCode[%s]\n",
            nDeviceID, nControlID, m_ctlXnsDevice.GetErrorString(nErrorCode));
    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnAlarmOutControlFailed

Description

Alarm-out 제어를 실패했을 때 발생하는 이벤트입니다.

Syntax

```
void OnAlarmOutControlFailed (
    long nDeviceID,
    long nControlID,
    long nErrorCode
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nErrorCode

[in] 에러 코드.

Example

[C++]

```
void CTestSampleDlg::OnAlarmOutControlFailedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nErrorCode)
{
    CString str;
    str.Format("AlarmOut Control Failed DeviceID[%d] ControlID[%d], Status[%d]\n",
nDeviceID, nControlID, m_ctlXnsDevice.GetErrorString(nErrorCode));
    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnControlRemoteNameChanged

Description

DVR 내의 control module의 이름이 변경되었을 때 발생합니다.

Syntax

```
void OnControlRemoteNameChanged(
    long nDeviceID,
    long nControlID
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

Example

[C++]

```
void CTestSampleDlg::OnControlRemoteNameChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID)
{
    CString str;
    str.Format("Control RemoteName Changed DeviceID[%d] ControlID[%d]\n", nDeviceID,
nControlID);
    OutputDebugString(str);
}
```

See Also

[ControlID](#)

OnGetPtzPos

Description

이 이벤트는 GetPtzPos()가 호출되면 발생하며, 이벤트 핸들러의 파라미터로 프리셋 리스트의 핸들이 전달됩니다. 이 핸들은 카메라의 절대좌표 정보를 얻는데 사용됩니다.

Syntax

```
void OnGetPtzPos(
    long nDeviceID,
    long nControlID,
    long nErrorCode,
    long nPan,
    long nTilt,
    long nZoom
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nErrorCode

[in] 리턴값

nPan

[in] Pan 절대 좌표값

nTilt

[in] Tilt 절대 좌표값

nZoom

[in] Zoom 절대 좌표값

Example

[C++]

```
void CMy01_LiveDlg::OnGetPtzPosXnssdkdevicectrl(long nDeviceID, long nControlID, long
nErrorCode, long nPan, long nTilt, long nZoom)
{
    TRACE(_T("Get Ptz Position pan=%d, tilt=%d, zoom=%d, ret=%d\n"),
          nPan, nTilt, nZoom, nErrorCode);
}
```

See Also

[GetPtzPos](#), [SetPtzPos](#), [ControlID](#)

OnGetPtzPosNormalize

Description

이 이벤트는 GetPtzPosNormalize()가 호출되면 발생하며, 이벤트 핸들러의 파라미터로 프리셋 리스트의 핸들이 전달됩니다. 이 핸들은 카메라의 좌표 정보를 얻는데 사용됩니다.

Syntax

```
void OnGetPtzPosNormalize(
    long nDeviceID,
    long nControlID,
    long nErrorCode,
    float fPan,
    float fTilt,
    float fZoom
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nErrorCode

[in] 리턴값

fPan

[in] Pan 좌표값

fTilt

[in] Tilt 좌표값

fZoom

[in] Zoom 좌표값

Example

[C++]

```
void CMy01_LiveDlg::OnGetPtzPosNormalizeXnssdkdevicectrl(long nDeviceID, long nControlID,
long nErrorCode, float fPan, float fTilt, float fZoom)
{
    TRACE(_T("Get Ptz Position Normalize pan=%f, tilt=%f, zoom=%f, ret=%d\n"),
          fPan, fTilt, fZoom, nErrorCode);
}
```

See Also

[GetPtzPosNormalize](#), [SetPtzPosNormalize](#), [ControlID](#)

OnAreaZoomMoving

Description

이 이벤트는 장비의 AreaZoom 이동 상태가 변경되었을 때 발생합니다. AreaZoom을 지원하는 장비에서 유효합니다.

Syntax

```
void OnAreaZoomMoving(
    long nDeviceID,
    long nControlID,
```

```
    long nStatus,
    long nReserved
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nStatus

[in] AreaZoom 이동 상태.

- 1: 이동 중

- 2: 이동 멈춤

nReserved

[in] 사용하지 않음. 0으로 설정

Example

[C++]

```
void CMy01_LiveDlg::OnAreaZoomMovingXnssdkdevicectrl(long nDeviceID, long nControlID,
long nStatus, long nReserved)
{
    if( bAreaZoom == True)
    {
        CString str;
        str.Format("AreaZoomMoving Changed DeviceID[%d] ControlID[%d],
Status[%d]\n", nDeviceID, nControlID, nStatus);
        OutputDebugString(str);
    }
}
```

See Also

[AreaZoom](#), [ControlID](#)

OnVideoFrameInfoReceived

Description

이 이벤트는 장비의 비디오 프레임 정보를 수신하였을 때 발생합니다. [SetVideoInfoReceived\(\)](#)를 통해 비디오 정보 수신을 활성화해야 합니다.

Syntax

```
void OnVideoFrameInfoReceived(
    long nDeviceID,
    long nControlID,
    long nMediaType,
    long nFps,
    long nKbps,
    long nVideoWidth,
    long nVideoHeight,
    long nCodecType
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nMediaType

[in] 미디어 타입.

- XMEDIA_LIVE (=1): Live

- XMEDIA_PLAYBACK (=2): Playback

- XMEDIA_BACKUP (=3): Backup replay (local PC에 스트림 데이터를 저장함)

nFps

[in] fps 정보

nKbps

[in] Bitrate 정보 (Kbps)

nVideoWidth

[in] 비디오 해상도 정보 (width)

nVideoHeight

[in] 비디오 해상도 정보 (height)

nCodecType

[in] 비디오 코덱정보

- XVCODEC_UNKNOWN (=0): Unknown
- XVCODEC_MJPEG (=1): MJPEG
- XVCODEC_MPEG4 (=2): MPEG4
- XVCODEC_H264 (=3): H264

Example

[C++]

```
void CMy01_LiveDlg::OnVideoFrameInfoReceived (long nDeviceID,
                                             long nControlID,
                                             long nMediaType,
                                             long nFps,
                                             long nKbps,
                                             long nVideoWidth,
                                             long nVideoHeight,
                                             long nCodecType)
{
    CString str;
    str.Format("Video frame received DeviceID[%d] ControlID[%d], fps[%d], Kbps[%d]\n",
               nDeviceID, nControlID, nFps, nKbps);
    OutputDebugString(str);
}
```

See Also

[SetVideoFrameInfoReceived](#), [ControlID](#)

OnRecordingErrorEvent

Description

이 이벤트는 장비에서 녹화 오류 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnRecordingErrorEvent(
    long nDeviceID,
    long tEventUTC
);
```

Parameters

nDeviceID

[in] 장비 ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

Example

[C++]

```
void CTestSampleDlg::OnRecordingErrorEventXnssdkdevicectrl(long nDeviceID, long tEventUTC)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("Recording Error Event DeviceID[%d], time[%d-%d-%d %02d:%02d:%02d]\n",
nDeviceID, Year, Mon, Day, Hour, Min, Sec);
    OutputDebugString(str);
}
```

OnPowerRestart

Description

이 이벤트는 애플리케이션이 PowerRestart()를 호출했을 때 발생합니다.

Syntax

```
void OnPowerRestart(
    long nDeviceID,
    long nErrorCode
);
```

Parameters

nDeviceID

[in] 장비 ID.

nErrorCode

[in] 에러 코드

Example

[C++]

```
void CMy01_LiveDlg::OnPowerRestartXnssdkdevicectrl(long nDeviceID, long nErrorCode)
{
    char str[100];
    sprintf(str, "Power Restart.. nDeviceID[%d], Error[%s]\n", nDeviceID,
m_ctlXnsDevice.GetErrorString(nErrorCode));

    OutputDebugString(str);}

```

See Also[PowerRestart](#)

OnRecordingStatusChanged

Description

이 이벤트는 장비의 녹화상태가 변경되었을 때, 발생하는 이벤트입니다.

Syntax

```
void OnRecordingStatusChanged(
    long nDeviceID,
    long tEventUTC,
    long nControlID,
    long nStatus
);
```

Parameters*nDeviceID*

[in] 장비 ID.

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임.)

nControlID

[in] Control ID.

nStatus

[in] 장비의 변경된 녹화상태를 나타냄.

- (1): 녹화 종료.

- (0): 녹화 시작.

Example

[C++]

```
void CTestSampleDlg::OnRecordingStatusChangedXnssdkdevicectrl(long nDeviceID, long
tEventUTC, long nControlID, long nStatus)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("RecordingStatusChanged Event DeviceID[%d], time[%d-%d-
%d %02d:%02d:%02d], ControlID[%d], Status[%d]\n", nDeviceID, Year, Mon, Day, Hour, Min,
Sec, nControlID, nStatus);
    OutputDebugString(str);
}
```

See Also

[ToggleDvrRecording](#)

OnTrackingEventChanged

Description

이 이벤트는 장비에서 트래킹 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnTrackingEventChanged(
    long nDeviceID,
    long nControlID,
    long nSubEvent,
    long tEventUTC,
    long nParam1,
    long nParam2,
    long nParam3,
    long nParam4,
    LPCTSTR description,
    LPCTSTR strParam
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nSubEvent

[in] Tracking Event의 세부 정보를 나타냄.

- XEVN_TRACKING_OCCUR (=1) : Tracking event occurred

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임)

nParam1

[in] (사용하지 않음).

nParam2

[in] (사용하지 않음).

nParam3

[in] (사용하지 않음).

nParam4

[in] (사용하지 않음).

description

[in] (사용하지 않음).

strParam

[in] (사용하지 않음).

Example

[C++]

```
void CTestSampleDlg::OnTrackingEventChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nSubEvent, long tEventUTC, long nParam1, long nParam2, long nParam3, long
nParam4, LPCTSTR description, LPCTSTR strParam)
{
    // TODO: Add your message handler code here
    long year, mon, day, hour, min, sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &year, &mon, &day, &hour, &min, &sec);
    DBG_LOG(_T("$$Tracking Event Detected, device=%d, cid=%d, time=[%04d-%02d-
%02d %02d:%02d:%02d]##n"),
            nDeviceID, nControlID, year, mon, day, hour, min, sec);
    if(nSubEvent & XEVN_TRACKING_OCCUR)
        DBG_LOG(_T("$$SUBEVENT TYPE : OCCUR##n"));
}
```

OnFaceDetectionEventChanged

Description

이 이벤트는 장비에서 face detection 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnFaceDetectionEventChanged(
    long nDeviceID,
    long nControlID,
    long nSubEvent,
    long tEventUTC,
    long nParam1,
    long nParam2,
    long nParam3,
    long nParam4,
    LPCTSTR description,
    LPCTSTR strParam
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nSubEvent

[in] Face Detection Event의 세부 정보를 나타냄.

- XEVT_FACE_DETECTION_OCCUR (=4) : Face detection event occurred

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임)

nParam1

[in] (사용하지 않음).

nParam2

[in] (사용하지 않음).

nParam3

[in] (사용하지 않음).

nParam4

[in] (사용하지 않음).

description

[in] (사용하지 않음).

strParam

[in] (사용하지 않음).

Example

[C++]

```
void CTestSampleDlg::OnFaceDetectionEventChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nSubEvent, long tEventUTC, long nParam1, long nParam2, long nParam3, long
nParam4, LPCTSTR description, LPCTSTR strParam)
{
    // TODO: Add your message handler code here
    long year, mon, day, hour, min, sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &year, &mon, &day, &hour, &min, &sec);
    DBG_LOG(_T("$$Face Detection Event Detected, device=%d, cid=%d, time=[%04d-%02d-
%02d %02d:%02d:%02d]₩n"),
            nDeviceID, nControlID, year, mon, day, hour, min, sec);
    if(nSubEvent & XEVN_FACE_DETECTION_OCCUR)
        DBG_LOG(_T("$$SUBEVENT TYPE : OCCUR₩n"));
}
```

OnAudioDetectionEventChanged

Description

이 이벤트는 장비에서 Audio detection 이벤트가 발생했음을 알려줍니다.

Syntax

```
void OnAudioDetectionEventChanged(
    long nDeviceID,
    long nControlID,
    long nSubEvent,
    long tEventUTC,
    long nParam1,
    long nParam2,
    long nParam3,
```

```

        long nParam4,
        LPCTSTR description,
        LPCTSTR strParam
);

```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nSubEvent

[in] Audio Detection Event의 세부 정보를 나타냄.

- XEVT_AUDIO_DETECTION_OCCUR (=7) : Audio detection event occurred

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임)

nParam1

[in] (사용하지 않음).

nParam2

[in] (사용하지 않음).

nParam3

[in] (사용하지 않음).

nParam4

[in] (사용하지 않음).

description

[in] (사용하지 않음).

strParam

[in] (사용하지 않음).

Example

[C++]

```

void CTestSampleDlg::OnAudioDetectionEventChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nSubEvent, long tEventUTC, long nParam1, long nParam2, long nParam3, long
nParam4, LPCTSTR description, LPCTSTR strParam)
{
    // TODO: Add your message handler code here
    long year, mon, day, hour, min, sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &year, &mon, &day, &hour, &min, &sec);
}

```

```

    DBG_LOG(_T("$$Audio Detection Event, device=%d, cid=%d, time=[%04d-%02d-
%02d %02d:%02d:%02d]Wn"),
    nDeviceID, nControlID, year, mon, day, hour, min, sec);
    if(nSubEvent & XEVN_AUDIO_DETECTION_OCCUR)
        DBG_LOG(_T("$$SUBEVENT TYPE : OCCURWn"));
}

```

OnAMDEventChanged

Description

이 이벤트는 장비에서 AMD(Advanced Motion Detected) 관련 이벤트가 발생했음을 알려줍니다.

Syntax

```

void OnAMDEventChanged(
    long nDeviceID,
    long nControlID,
    long nSubEvent,
    long tEventUTC,
    long nParam1,
    long nParam2,
    long nParam3,
    long nParam4,
    LPCTSTR description,
    LPCTSTR strParam
);

```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nSubEvent

[in] AMD Event의 세부 정보를 나타냄. ([표 2.12](#) 참조)

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임)

nParam1

[in] (사용하지 않음).

nParam2

[in] (사용하지 않음).

nParam3

[in] (사용하지 않음).

nParam4

[in] (사용하지 않음).

description

[in] (사용하지 않음).

strParam

[in] (사용하지 않음).

표 2.14 AMD SubEvent List

SubEvent Type	Value	Description
XEVN_AMD_ROI_MOTION_DETECTION_START	10	ROI 영역에서 모션 검출 시작
XEVN_AMD_ROI_MOTION_DETECTION_STOP	11	ROI 영역에서 모션 검출 종료
XEVN_AMD_LIB_LOAD_FAILED	12	AMD 라이브러리 로드 실패
XEVN_AMD_LOW_FPS_START	13	fps 가 10 미만일 경우
XEVN_AMD_LOW_FPS_END	14	fps 가 10 이상일 경우
XEVN_AMD_DSP_DISPLAY_START	15	DSP(DM8167) Display 시작
XEVN_AMD_DSP_DISPLAY_FAIL	16	DSP(DM8167) Display 실패
XEVN_AMD_DSP_VA_START	17	DSP(DM8167) VA 시작
XEVN_AMD_DSP_VA_FAIL	18	DSP(DM8167) VA 실패

Example

[C++]

```
void CTestSampleDlg::OnAMDEventChangedXnssdkdevicectrl1(long nDeviceID, long nControlID,
long nSubEvent, long tEventUTC, long nParam1, long nParam2, long nParam3, long nParam4,
LPCTSTR description, LPCTSTR strParam)
{
    // TODO: Add your message handler code here
    long year, mon, day, hour, min, sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &year, &mon, &day, &hour, &min, &sec);
    DBG_LOG_T("$$AMD Event Detected, device=%d, cid=%d, time=[%04d-%02d-
%02d %02d:%02d:%02d]\n",
```

```

    nDeviceID, nControlID, year, mon, day, hour, min, sec);
    if(nSubEvent & XEVT_AMD_ROI_MOTION_DETECTION_START)
        DBG_LOG(_T("$$SUBEVENT TYPE : ROI_MOTION_DETECTION_START\n"));
    else if(nSubEvent & XEVT_AMD_LOW_FPS_START)
        DBG_LOG(_T("$$SUBEVENT TYPE : AMD_LOW_FPS_START\n"));
}

```

OnSystemEventChanged

Description

이 이벤트는 장비에서 system 이벤트가 발생했음을 알려줍니다.

Syntax

```

void OnSystemEventChanged(
    long nDeviceID,
    long nControlID,
    long nSubEvent,
    long tEventUTC,
    long nParam1,
    long nParam2,
    long nParam3,
    long nParam4,
    LPCTSTR description,
    LPCTSTR strParam
);

```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nSubEvent

[in] System Event의 세부 정보를 나타냄. (표 2.12 참조)

tEventUTC

[in] 이벤트 발생 시간. (4바이트의 time_t 포맷임)

nParam1

[in] 일부 nSubEvent의 세부 정보를 나타냄. (표 2.12 참조)

nParam2

[in] (사용하지 않음).

nParam3

[in] (사용하지 않음).

nParam4

[in] (사용하지 않음).

description

[in] (사용하지 않음).

strParam

[in] (사용하지 않음).

표 2.15 System SubEvent List

SubEvent Type	Value	Description
XEVN_PROFILE_ADD	19	카메라 프로파일 추가
XEVN_PROFILE_CHANGED	20	카메라 프로파일 설정 변경
XEVN_PROFILE_REMOVED	21	카메라 프로파일 제거
XEVN_BEING_UPDATED	22	NVR 장비 펌웨어가 업데이트 되고 있음. 업데이트 시작시 발생
XEVN_RECORD_FILTERING_START	23	(NVR)레코딩 중 프레임 스kip 발생
XEVN_RECORD_FILTERING_STOP	24	(NVR)레코딩 중 프레임 스kip 발생 종료
XEVN_BATTERY_FAIL	25	(NVR)RTC(Real Time Clock) battery 방전
XEVN_NETWORK_TRAFFIC_STATUS_OVERFLOW	26	카메라로부터 받은 데이터 수신량이 NVR 최대 수신가능 데이터량 초과
XEVN_NETWORK_TRAFFIC_STATUS_RELEASE	27	카메라로부터 받은 데이터 수신량이 NVR 최대 수신가능 데이터량 초과 해제
XEVN_UPGRADE_LOG	28	(NVR)업데이트 서버에 업그레이드 할 새로운 버전의 펌웨어가 있음
XEVN_OVERWRITE_DECODING	29	(NVR)Search 재생중인 구간에 overwrite 발생
XEVN_RAID_MODE_ENABLE	30	(NVR)RAID 모드 변경시에 발생 • nParam1 : RAID Array Number(1 또는 2).

XEVN_RAID_MODE_DISABLE	31	(NVR)RAID 모드 해제
XEVN_RAID_SETUP_CHANGE	32	(NVR)RAID 설정 변경
XEVN_RAID_BUILD_CANCEL	33	(NVR)RAID 빌드 취소
XEVN_RAID_BUILD_FAIL	34	(NVR)RAID 빌드중 에러 발생
XEVN_RAID_REBUILDING_START	35	(NVR)RAID device 복구 시작 (고장난 디바이스를 대체한 새로운 디바이스에 복구)
XEVN_RAID_REBUILDING_END	36	(NVR)RAID device 복구 종료
XEVN_RAID_REBUILDING_FAIL	37	(NVR)RAID device 복구 실패
XEVN_RAID_DEGRADED	38	(NVR)RAID device 고장(고장시 복구되지 않음)
XEVN_RAID_FAIL	39	(NVR)RAID array 를 사용할 수 없는 상황
XEVN_ISCSI_CONNECTED	40	(NVR)iSCSI 장비 연결 완료
XEVN_ISCSI_DISCONNECTED	41	(NVR)iSCSI 장비 연결 해제
XEVN_RECORDING_FRAME_DROP	42	(NVR)성능 문제로 인한 영상 손실
XEVN_DUAL_SMPS_FAILURE	43	(NVR)Dual SMPS 오류
XEVN_DUAL_SMPS_RECOVERED	44	(NVR)Dual SMPS 복구
XEVN_INTERNAL_HDD_CONNECTED	45	(NVR)Internal HDD 추가시 발생 • nParam1 : HDD number
XEVN_INTERNAL_HDD_DISCONNECTED	46	(NVR)Internal HDD 제거시 발생 • nParam1 : HDD number
XEVN_USB_HDD_CONNECTED	47	(NVR)USB HDD 추가시 발생
XEVN_USB_HDD_DISCONNECTED	48	(NVR)USB HDD 제거시 발생
XEVN_BACKUP_START	49	(NVR)Backup 시작
XEVN_BACKUP_END	50	(NVR)Backup 종료
XEVN_ALARM_RESET	51	(NVR)장비에 표시된 모든 알람 리셋 Alram Reset 이 되었을 때 발생
XEVN_RAID_DEVICE_ADD	52	(NVR)Hot Swap RAID 추가
XEVN_RAID_DEVICE_ADD_FAIL	53	(NVR)Hot Swap RAID 추가 실패
XEVN_RAID_REC_RESTRICTION	54	(NVR)RAID 빌드시 녹화 제한하는 경우 발생

Example

[C++]

```

void CTestSampleDlg::OnSystemEventChangedXnssdkdevicectrl1(long nDeviceID, long
nControlID, long nSubEvent, long tEventUTC, long nParam1, long nParam2, long nParam3, long
nParam4, LPCTSTR description, LPCTSTR strParam)
{
    // TODO: Add your message handler code here
    long year, mon, day, hour, min, sec;
    m_ctlXnsDevice.TimetToDate(tEventUTC, &year, &mon, &day, &hour, &min, &sec);
    DBG_LOG(_T("$$System Event Detected, device=%d, cid=%d, time=[%04d-%02d-
%02d %02d:%02d:%02d]₩n"),
            nDeviceID, nControlID, year, mon, day, hour, min, sec);
    if(nSubEvent & XEVN_PROFILE_ADD)
        DBG_LOG(_T("$$SUBEVENT TYPE : PROFILE_ADD₩n"));
    if(nSubEvent & XEVN_PROFILE_CHANGED)
        DBG_LOG(_T("$$SUBEVENT TYPE : PROFILE_CHANGED₩n"));
}

```

OnChangeDeviceAdminPassword

Description

이 이벤트는 애플리케이션이 ChangeDeviceAdminPassword()를 이용하여 Admin계정의 비밀번호를 변경했을 때 발생하며, 함수 수행 결과를 포함하고 있습니다.

Syntax

```

void OnChangeDeviceAdminPassword(
    long hDevice,
    long nErrorCode
);

```

Parameters

hDevice

[in] Device handle. 이 값은 CreateDevice()에서 반환한 핸들임.

nErrorCode

[in] 에러코드

Example

[C++]

```
Void CSdkTestDlg::OnChangeDeviceAdminPasswordXnssdkdevicectrl(long nDeviceID, long  
nErrorCode)  
{  
    WLOGD(_T("Change Password, errcode=[%d](%s)\n"), nErrorCode,  
m_ctrlXnsDevice.GetErrorString(nErrorCode));  
    if(nErrorCode == ERR_SUCCESS)  
    {  
        m_pDlgDeviceInfo->ReconnectNewPw(m_pChangePw->m_strNewPw);  
    }  
}
```

See Also

[ChangeDeviceAdminPassword](#)

CHAPTER 3

XnsSdkWindow

XnsSdkWindow는 미디어를 재생할 수 있는 원도우를 제공하고 화면에 표시되는 정보들을 관리하는 컴포넌트입니다.

이 컴포넌트에서 제공하는 기능은 다음과 같습니다.

- 미디어 정보 관리
- 줌(zoom) 제어
- OSD 메뉴
- 볼륨(volume) 제어
- 미디어 재생
- 버퍼 관리
- 화면에 표시되는 정보 관리
- 마우스와 키보드 입력 처리

Contents

Initialize the Control

Start and Stop

Get Device Information

Control Zoom

OSD Menu

Control Volume

Snapshot

Control Playback

Control Buffer

Get Display Information

Miscellaneous

Events

Initialize the Control

XnsSdkWindow.ocx는 Initialize()를 이용하여 초기화를 해야 사용할 수 있습니다. 그러나 사용을 마친 후에는 종료 함수를 호출하지 않아도 됩니다.

Initialize

Description

XnsSdkWindow 컨트롤을 초기화합니다. 즉, 이미지를 화면에 출력하기 위해 윈도우 핸들(window handle)을 설정합니다.

Syntax

```
long Initialize(
    long hDrawWnd,
    long hTopWnd
);
```

Parameters

hDrawWnd

[in] 윈도우 핸들. 이 값이 NULL이면, XnsSdkWindow는 기본으로 설정된 윈도우(default window)를 사용함.

hTopWnd

[in] 부모 윈도우의 핸들. 이 값이 NULL이면, XnsSdkWindow는 기본으로 설정된 윈도우(default window)를 사용함.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.

Example

[C++]

```
long ErrCode = m_ctlXnsWindow.Initialize(NULL, NULL);
char str[100];
sprintf(str, "ErrorCode[0x%lx]\n", ErrCode);
OutputDebugString(str);
```

See Also

[XnsSdkDevice::Initialize](#)

Start and Stop

Start

Description

Media source 핸들을 XnsSdkWindow에 등록하는 함수입니다.

Media source 핸들은 XnsSdkDevice에서 생성되는 핸들로, 애플리케이션이 XnsSdkDevice::OpenMedia()를 호출하면 장비로부터 미디어 스트림이 수신되고 그와 함께 Media Source 핸들이 반환됩니다. 애플리케이션은 Start()를 이용해서 Media source 핸들을 XnsSdkWindow로 전달해야 XnsSdkWindow가 스트림 데이터를 얻을 수 있습니다.

Syntax

```
long Start(long hMediaSource);
```

Parameters

hMediaSource

[in] Media source의 핸들. 이 값은 XnsSdkDevice::[OpenMedia\(\)](#)를 이용해서 얻을 수 있음.

주) OpenStream()을 사용할 경우 hMediaSource 값은 0을 사용합니다.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.
- ERR_ALREADY_RUN: 이미 실행 중임.

Example

[C++]

```
m_ctlXnsDevice.OpenMedia(m_hDevice, 2, XMEDIA_LIVE, 0, 0, &m_pMediaLiveSource);
m_ctlXnsWindow.Start(m_pMediaLiveSource);
```

See Also

[XnsSdkDevice::OpenMedia](#), [Stop](#)

Stop

Description

Media source 핸들(*pMediaSource*)로부터 스트림 데이터 수신을 중단합니다.

Syntax

```
long Stop(void);
```

Return Value

Media source의 핸들.

Example

[C++]

```
m_ctlXnsWindow.Stop();
m_ctlXnsDevice.CloseMedia(m_hDevice, m_pMediaLiveSource);
m_pMediaLiveSource = NULL;
```

See Also

[XnsSdkDevice::CloseMedia](#), [Start](#)

Get Device Information

GetDeviceID

Description

미디어 소스(media source)와 연결된 장비의 ID를 반환합니다.

Device ID는 애플리케이션에서 XnsSdkDevice::CreateDevice()를 이용하여 정의한 값입니다.

Syntax

```
long GetDeviceID(void);
```

Return Value

Device ID. 리턴 값이 0이면, 매칭되는 device ID가 없음을 의미합니다.

Example

[C++]

```
long nDeviceID = m_ctlXnsWindow.GetDeviceID();
DisplayLong("GetDeviceID", nDeviceID);
```

See Also

[XnsSdkDevice::CreateDevice](#)

GetControlID

Description

미디어 소스(media source)로부터 Control ID를 반환합니다.

Syntax

```
long GetControlID(void);
```

Return Value

미디어 소스와 연결된 Control ID.

Example

[C++]

```
long nControlID = m_ctlXnsWindow.GetControlID();
DisplayLong("GetControlID", nControlID);
```

GetImageWidth

Description

원본 이미지(source image)의 너비를 반환합니다.

Syntax

```
long GetImageWidth(void);
```

Return Value

원본 이미지의 너비.

Example

[C++]

```
long nWidth = m_ctlXnsWindow.GetImageWidth();
DisplayLong("ImageWidth", nWidth);
```

See Also

[GetImageHeight](#)

GetImageHeight

Description

원본 이미지의 높이를 반환합니다.

Syntax

```
long GetImageHeight(void);
```

Return Value

원본 이미지의 높이.

Example

[C++]

```
long nHeight = m_ctlXnsWindow.GetImageHeight ();
DisplayLong("ImageHeight", nHeight);
```

See Also

[GetImageWidth](#)

IsMedia

Description

스트림 소스가 있는지 여부를 체크합니다.

Syntax

```
long IsMedia(void);
```

Return Value

다음 조건을 만족하는 경우 1을 반환하고, 그렇지 않을 경우 0을 반환합니다.

- XnsSdkWindow가 초기화됨.
- hMediaSource가 유효함.
- XnsSdkWindow에 설정된 장비가 애플리케이션과 연결되어 있고, 스트림 데이터를 전송함.

Example

[C++]

```
long isMedia = m_ctlXnsWindow.IsMedia();
DisplayLong("IsMedia", isMedia);
```

See Also

[Start](#)

GetMediaType

Description

출력하고 있는 미디어의 타입을 반환합니다.

Syntax

```
long GetMediaType(void);
```

Return Value

미디어 타입.

- MEDIA_TYPE_LIVE(1): 실시간 스트리밍. (Live streaming)
- MEDIA_TYPE_PLAYBACK(2): 원격에 저장된 데이터를 재생함.
- MEDIA_TYPE_BACKUP(3): 로컬 PC에 저장된 데이터를 재생함.

Example

[C++]

```
long MediaType = m_ctlXnsWindow.GetMediaType();
DisplayLong("GetMediaType", MediaType);
```

Control Zoom

ZoomLevel

Description

디지털 줌을 제어합니다.

Syntax

```
long ZoomLevel(
    long nX,
    long nY,
    long nLevel
);
```

Parameters

nX

[in] 중점으로 설정할 X축 좌표.

nY

[in] 중점으로 설정할 Y축 좌표.

nLevel

[in] Zoom-in/out의 비율(%). 이때, 양수는 zoom-in, 음수는 zoom-out을 의미함. 이 값이 100인 경우, 1배율 (full image)이며, 200인 경우 2x로 zoom-in함. (즉, 200% zoom-in) 최대값은 10000임.

참고

이미지의 좌측 상단은 (0,0), 우측 하단은 (width, height)입니다.

Return Value

줌 비율(%). IsMedia()의 반환 값이 FALSE인 경우, 이 함수의 리턴 값은 100으로 고정됩니다.

Example

[C++]

```
long nZoomLevel = m_ctlXnsWindow.ZoomLevel(50, 50, 400);
DisplayLong("ZoomLevel", nZoomLevel);
```

See Also

[IsMedia](#)

ZoomRate

Description

줌 비율(zoom-ratio)를 이용하여 디지털 줌을 제어합니다.

Syntax

```
long ZoomRate(
    double lfX,
    double lfY,
    double lfWidth,
    double lfHeight
);
```

Parameters

lfX

[in] Zoom ratio를 사용하여 좌측 상단의 X좌표를 계산한 값.

lfY

[in] Zoom ratio를 사용하여 좌측 상단의 Y좌표를 계산한 값.

lfWidth

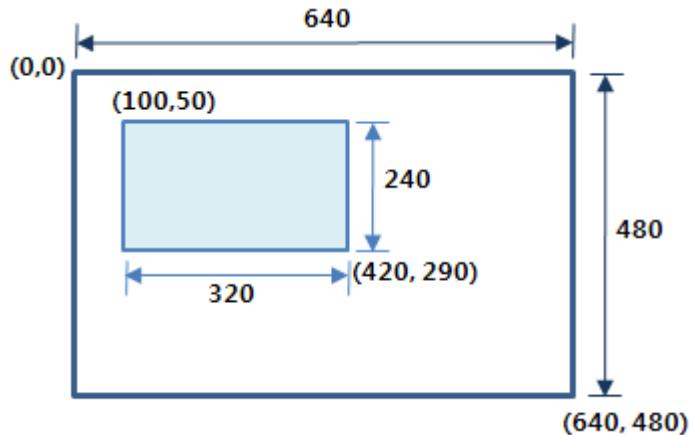
[in] Zoom ratio를 사용하여 너비를 계산한 값.

lfHeight

[in] Zoom ratio를 사용하여 높이를 계산한 값.

예를 들어, XnsSdkWindow 컨트롤의 크기가 640 x 480이고, 줌 영역(zoom rectangle)의 사이즈가 320 x 240, 좌측 상단의 좌표가 (100, 50)인 경우, 위의 파라미터는 다음과 같이 계산됩니다.

그림 3.1 Zoom Rectangle의 예



IfX = 100/640 = 0.15625

IfY = 50/480 = 0.1041667

IfWidth = 320/640 = 0.5

IfHeight = 240/480 = 0.5

Return Value

줌 비율(%). IsMedia()가 FALSE를 리턴하는 경우, 이 함수의 리턴 값은 100으로 고정됩니다.

Example

[C++]

```
long ZoomRate = m_ctlXnsWindow.ZoomRate(.25, .25, .5, .5);
DisplayLong("ZoomRate", ZoomRate);
```

See Also

[ZoomLevel](#), [ClearZoom](#)

GetZoomInfo

Description

현재 설정된 디지털 줌의 비율(ratio)를 반환합니다.

Syntax

```
long GetZoomInfo(
    double* plfX,
    double* plfY,
    double* plfWidth,
```

```
    double* plfHeight  
);
```

Parameters

plfX

[out] Zoom ratio를 사용하여 좌측 상단의 X좌표를 계산한 값.

plfY

[out] Zoom ratio를 사용하여 좌측 상단의 Y좌표를 계산한 값.

plfWidth

[out] Zoom ratio를 사용하여 너비를 계산한 값.

plfHeight

[out] Zoom ratio를 사용하여 높이를 계산한 값.

참고

Zoom ratio를 계산하는 방법은 ZoomRate의 예제를 참고하시기 바랍니다.

Return Value

줌 비율(%).

참고

IsMedia()가 FALSE를 리턴하는 경우, 이 함수의 리턴 값은 100으로 고정됩니다.

Example

[C++]

```
double lfX, lfY, lfWidth, lfHeight;  
long nZoomFactor = m_ctlXnsWindow.GetZoomInfo(&lfX, &lfY, &lfWidth, &lfHeight);  
  
CString str;  
str.Format("GetZoomInfo.. lfX[%f], lfY[%f], lfWidth[%f], lfHeight[%f]\n", lfX, lfY, lfWidth, lfHeight);  
OutputDebugString(str);
```

See Also

[ZoomRate](#)

ClearZoom

Description

줌 설정을 삭제합니다. 이미지를 원본 크기로 복원합니다.

Syntax

```
void ClearZoom(void);
```

Example

[C++]

```
m_ctlXnsWindow.ClearZoom();
```

See Also

[ZoomLevel](#), [ZoomRate](#)

DrawRect

Description

파란색 브러쉬로 XnsSdkWindow에 사각형을 그립니다. 이 함수가 두 번 이상 불리게 되면, 이전에 그려진 사각형은 지워지고, 새로운 사각형이 다시 그려집니다.

Syntax

```
void DrawRect(
    long nX,
    long nY,
    long nWidth,
    long nHeight
);
```

Parameters

nX

[in] 좌측 상단의 X 좌표.

nY

[in] 좌측 상단의 Y 좌표.

nWidth

[in] 사각형의 너비.

nHeight

[in] 사각형의 높이.

Example

[C++]

```
m_ctlXnsWindow.DrawRect(10, 10, 50, 50);
```

See Also

[ClearDraw](#), [DrawShape](#)

ClearDraw

Description

화면에 그려진 사각형을 지웁니다.

Syntax

```
void ClearDraw(void);
```

Example

[C++]

```
m_ctlXnsWindow.ClearDraw();
```

See Also

[DrawRect](#)

DrawShape

Description

애플리케이션에서 설정한 색상과 굵기로 사각형을 그립니다. 이 사각형을 지우려면, 다른 파라미터 값은 이전과 동일하게 설정하고 굵기(*nThick*)만 0으로 하여 이 함수를 다시 호출합니다.

Syntax

```
void DrawShape(
    long nIndex,
    long nType,
    long nX,
    long nY,
    long nWidth,
    long nHeight,
    long nR,
    long nG,
```

```
    long nB,
    long nThick
);
```

Parameters

nIndex

[in] 사각형의 인덱스. (1~10 사이의 값을 가짐.)

nType

[in] 명령 타입

Draw (=1): 사각형을 그림.

Erase (= -1): 사각형을 지움.

nX

[in] 좌측 상단의 X 좌표.

nY

[in] 좌측 상단의 Y 좌표.

nWidth

[in] 사각형의 너비.

nHeight

[in] 사각형의 높이.

nR

[in] RGB 색상 코드에서 Red 영역의 값.

nG

[in] RGB 색상 코드에서 Green 영역의 값.

nB

[in] RGB 색상 코드에서 Blue 영역의 값.

nThick

[in] 선의 굵기

Example

[C++]

```
m_ctlXnsWindow.DrawShape(1, 1, 10, 10, 50, 50, 255, 0, 0, 10);
```

See Also

[DrawRect](#)

OSD Menu

SetOSDOnOff

Description

영상 화면에 OSD 메뉴를 표시하거나 표시된 OSD 메뉴를 종료합니다.

Syntax

```
void SetOSDOnOff(  
    long nOSDID,  
    BOOL bOnOff  
)
```

Parameters

nOSDID

[in] OSD 항목의 ID.

bOnOff

[in] 메뉴의 on/off 플래그.

- On (=1): OSD 메뉴를 켭니다.
- Off (=0): OSD 메뉴를 끕니다.

표 3.1 OSD Type

Type	Value	Description
OSD_ALL	0xFFFF	모든 OSD 를 표시.
OSD_TIME1	0x0001	날짜와 시간. 둘째 줄에 표시함(default).
OSD_TIME2	0x0002	날짜와 시간. 첫째 줄에 표시함.
OSD_IPS	0x0004	초당 이미지 수. (Images per second)
OSD_CAMNAME	0x0008	카메라 이름.
OSD_SPEED	0x0010	재생 속도.
OSD_PTZ	0x0020	PTZ 제어 메뉴.

OSD_LISTEN	0x0040	Listen 기능 제어.
OSD_TALK	0x0080	Talk 상태를 화면에 표시함.
OSD_ALARM	0x0100	센서 이벤트.
OSD_MOTION	0x0200	모션 이벤트.
OSD_RECORDING	0x0400	로컬 PC에 녹화.

Example

[C++]

```
m_ctlXnsWindow.SetOsdOnOff(OSD_ALL, 0);
```

SetDateTimeFormat

Description

화면에 표시되는 날짜와 시간 정보의 포맷을 변경합니다.

Syntax

```
void SetDateTimeFormat(long nFormatID);
```

Parameters

nFormatID

[in] 날짜와 시간 포맷

- 1: YYYY-MM-DD hh:mm:ss
- 2: YYYY-MM-DD hh:mm
- 3: MM-DD-YYYY hh:mm:ss
- 4: MM-DD-YYYY hh:mm
- 5: DD-MM-YYYY hh:mm:ss
- 6: DD-MM-YYYY hh:mm

Remarks

윈도우의 너비가 너무 작아서 날짜와 시간을 다 표기하기 어려운 경우, XnsSdkWindow는 날짜 정보를 표기하지 않습니다.

Example

[C++]

```
m_ctlXnsWindow.SetDateTimeFormat(5);
```

SetCustomOSD

Description

영상 화면에 표시할 사용자 정의 OSD 정보를 설정합니다.

Syntax

```
long SetCustomOSD(
    BSTR szOSDText,
    long nX,
    long nY,
    long nR,
    long nG,
    long nB
);
```

Parameters

szOSDText

[in] 사용자 정의 문자열.

nX

[in] 좌측 상단의 X 좌표.

nY

[in] 좌측 상단의 Y 좌표.

nR

[in] RGB 색상 코드에서 Red 영역의 값.

nG

[in] RGB 색상 코드에서 Green 영역의 값.

nB

[in] RGB 색상 코드에서 Blue 영역의 값.

Example

[C++]

```
m_ctlXnsWindow.SetOSDText("Custom string", 10, 10, 255, 255, 255);
```

See Also

[SetCustomOSDOnOff](#)

SetCustomOSDOnOff

Description

영상 화면에 사용자 정의 OSD를 표시하거나 표시된 사용자 정의 OSD를 종료합니다.

Syntax

```
void SetOSDOnOff(
    BOOL bOnOff
);
```

Parameters

bOnOff

[in] 메뉴의 on/off 플래그.

- On (=1): 사용자 정의 OSD를 켕니다.
- Off (=0): 사용자 정의 OSD를 끕니다.

Example

[C++]

```
m_ctlXnsWindow.SetCustomOSDOnOff(1);
```

See Also

[SetCustomOSD](#)

Contorl Volume

SetSound

Description

사운드를 켜거나 끕니다.

Syntax

```
void SetSound(BOOL bSound);
```

Parameters

bSound

[in] 사운드의 on/off 플래그.

- 1: 사운드를 켭니다.
- 0: 사운드를 끕니다.

Example

[C++]

```
m_ctlXnsWindow.SetSound(0);
```

See Also

[SetListen](#), [SetVolume](#), [GetVolume](#)

SetVolume

Description

사운드의 볼륨을 설정합니다.

Syntax

```
void SetVolume(long nVolume);
```

Parameters

nVolume

[in] 사운드 볼륨. (0~100 사이의 값을 가짐.)

Example

[C++]

```
m_ctlXnsWindow.SetVolume(70);
```

See Also

[GetVolume](#), [SetSound](#)

GetVolume

Description

현재 설정된 사운드의 볼륨을 반환합니다.

Syntax

```
long GetVolume(void);
```

Return Value

현재의 사운드 볼륨. 이전에 설정된 값이 없다면, 기본으로 설정된 50이 리턴됩니다.

Example

[C++]

```
m_ctlXnsWindow.GetVolume();
```

See Also

[SetSound](#), [SetVolume](#)

Video Raw Data

SetVideoRawDataReceived

Description

디코딩된 비디오 데이터를 OnvideRawDataReceived() 이벤트를 통해 받고자 할 때 사용합니다.

Syntax

```
long SetVideoRawDataReceived(  
                           long hMediaSource  
)
```

Parameters

hMediaSource

[in] Media source의 핸들. 이 값은 XnsSdkDevice::[OpenMedia\(\)](#)를 이용해서 얻을 수 있음.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.

Example

[C++]

```
m_ctrlXnsSdkDevice.OpenMedia(m_hDevice, 3, 1, 0, 0, &m_hMediaSource);
m_ctrlXnsSdkWindow.Start(m_hMediaSource);
m_ctrlXnsSdkWindow.SetVideoRawDataReceived(m_hMediaSource);
```

See Also

[UnsetVideoRawDataReceived](#), [OnVideoRawDataReceived](#), [SaveSnapshotWithData](#), [OpenMedia](#), [Start](#)

UnSetVideoRawDataReceived

Description

디코딩된 비디오 데이터를 OnVideoRawDataReceived() 이벤트를 통해 받는 동작을 취소할 때 사용합니다.

Syntax

```
long UnSetVideoRawDataReceived(
    long hMediaSource
);
```

Parameters*hMediaSource*

[in] Media source의 핸들. 이 값은 XnsSdkDevice::[OpenMedia\(\)](#)를 이용해서 얻을 수 있음.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_NOT_INITIALIZE: 라이브러리가 초기화 되지 않음.

Example

[C++]

```
m_ctrlIXnsSdkDevice.OpenMedia(m_hDevice, 3, 1, 0, 0, &m_hMediaSource);
m_ctrlIXnsSdkWindow.Start(m_hMediaSource);
m_ctrlIXnsSdkWindow.SetVideoRawDataReceived(m_hMediaSource);
m_ctrlIXnsSdkWindow.UnSetVideoRawDataReceived(m_hMediaSource);
```

See Also

[SetVideoRawDataReceived](#), [OnVideoRawDataReceived](#), [SaveSnapshotWithData](#)

Snapshot

SaveSnapshot

Description

현재 이미지의 스냅샷(snapshot)을 저장합니다.

Syntax

```
long SaveSnapshot(
    BSTR szFile,
    long nFormat
);
```

Parameters

szFile

[in] 저장할 파일 이름.

nFormat

[in] 이미지 파일 포맷

- 0: BMP 파일

- 1: JPG 파일

Return Value

이미지 저장에 성공하면 1을, 실패하면 0을 리턴합니다.

Example

[C++]

```
long nSuccess= m_ctlXnsWindow.SaveSnapshot("d:\test_XnsWindow1.jpg", 1);
DisplayLong("SaveSnapshot", nSuccess);
```

See Also

[PrintSnapshot](#)

SaveSnapshotWithString

Description

현재 이미지의 스냅샷(snapshot)에 디바이스 정보를 표기하여 저장합니다.

Syntax

```
long SaveSnapshot(
    BSTR szFile,
    long nFormat,
    BOOL bShowCameraName,
    BOOL bUseLocalName,
    BOOL bShowDateTime,
    LPCTSTR szUserNote,
    BOOL bUseTempSavedImage
);
```

Parameters

szFile

[in] 저장할 파일 이름.

nFormat

[in] 이미지 파일 포맷

- 0: BMP 파일

- 1: JPG 파일

nShowCameraName

[in] snapshot에 camera name 출력을 지정하는 플래그.

- On (=1): camera name을 출력.

- Off (=0): camera name을 출력하지 않음.

nUseLocalName

[in] snapshot에 local name 출력을 지정하는 플래그

- On (=1): local name을 출력.
- Off (=0): local name을 출력하지 않음.

nShowDateTime

[in] snapshot에 날짜와 시간 출력을 지정하는 플래그

- On (=1): 날짜와 시간을 출력.
- Off (=0): 날짜와 시간을 출력하지 않음.

szUserNote

[in] snapshot에 사용자가 원하는 문구 출력

bUseTempSavedImage

[in] 사용 안함

Return Value

이미지 저장에 성공하면 1을, 실패하면 0을 리턴합니다.

Example**[C++]**

```
long nSuccess= m_ctlXnsWindow.SaveSnapshot("d:\WWtest_XnsWindow1.jpg", 1, 1, 0, 1, "Area :  
Floor1", 0 );  
DisplayLong("SaveSnapshot", nSuccess);
```

See Also

[PrintSnapshot](#)

SaveSnapshotWithData

Description

OnVideoRawDataReceived() 이벤트를 통해 받은 디코딩된 비디오데이터를 이미지 파일로 저장합니다.

Syntax

```
long SaveSnapshotWithData(  
                           long pData,
```

```

        long nDataSize,
        long nImageWidth,
        long nImageHeight,
        BSTR szFile,
        long nFormat
);

```

Parameters

pData

[in] 디코딩된 비디오데이터 버퍼

nDataSize

[in] 디코딩된 비디오데이터 바이트 크기

nImageWidth

[in] 디코딩된 비디오데이터 Width 해상도

nImageHeight

[in] 디코딩된 비디오데이터 Height 해상도

szFile

[in] 저장할 파일 이름.

nFormat

[in] 이미지 파일 포맷

- 0: BMP 파일

- 1: JPG 파일

Return Value

이미지 저장에 성공하면 1을, 실패하면 0을 리턴합니다.

Example

[C++]

```

void CVideoRawDataDlg::OnVideoRawDataReceivedXnssdkwindowctrl(long nDeviceID, long
nControlID, long nWidth, long nHeight, long nVideoSize, long pVideoData)
{
    m_nWidth = nWidth;
    m_nHeight = nHeight;
    unsigned char *pVideo = (unsigned char*)pVideoData;

    m_ctrlXnsSdkWindow.SaveSnapshotWithData(pVideoData, nVideoSize, nWidth, nHeight,
_T("C:\Temp\Samsung2.jpg"), 1);
}

```

```
}
```

See Also

[SetVideoRawDataReceived](#), [OnVideoRawDataReceived](#), [UnSetVideoRawDataReceived](#)

PrintSnapshot

Description

현재 이미지의 스냅샷(snapshot)을 프린터를 이용하여 출력합니다.

Syntax

```
long PrintSnapshot(void);
```

Return Value

이미지 출력에 성공하면 1을, 실패하면 0을 리턴합니다.

Example

[C++]

```
long nSuccess= m_ctlXnsWindow.PrintSnapshot();
DisplayLong("PrintSnapshot", nSuccess);
```

See Also

[SaveSnapshot](#), [SaveSnapshotWithString](#)

Control Playback

PushMedia

Description

윈도우에 미디어 스트림을 디코딩하여 화면에 출력하도록 XnsSdkWindow로 스트림을 보냅니다.

XnsSdkDevice::OpenStream()이 호출되면, 스트림 데이터는 OnNewMedia 이벤트를 통해 애플리케이션으로 전달됩니다. 애플리케이션은 PushMedia()를 이용하여 이 스트림을 XnsSdkWindow로 전달해야 영상이 디코딩되어 화면에 출력됩니다. 이 함수는 OnNewMedia 이벤트의 이벤트 핸들러 내에서만 올바르게 동작합니다.

Syntax

```
long PushMedia(long hMediaData);
```

Parameters

hMediaData

[in] 미디어 데이터. 이 값은 OnNewMedia 이벤트 핸들러의 파라미터로 전달되며, 이벤트 핸들러 함수 내에서만 유효함.

Return Value

- ERR_SUCCESS: 함수 수행에 성공함.
- ERR_BUFFER_FULL: 버퍼에 여유 공간이 없음.
- ERR_INVALID_SEQUENCE: 유효하지 않은 시퀀스(Sequence)임.

Remarks

이 함수를 사용하려면 사전에 Start()가 호출되어야 합니다.

Start()의 파라미터는 0을 입력하여야 합니다.

Example

[C++]

```
// PushMedia()가 호출되기 전에 Start()가 먼저 호출되어야 합니다.

void CTestSampleDlg::PushMedia()
{
    m_pMediaSource = 0;
    m_ctlXnsWindow.Start(m_pMediaSource);
    m_bPushMedia = 1;
}

void CTestSampleDlg::OnNewMediaXnssdkdevicectrl1(long nDeviceID, long nControlID, long
nMediaType, long hMediaData, long nInputType, long nFrameType, long nFrameTime, long
nFrameSize, long nTimeStamp)
{
    if (m_bPushMedia)
    {
```

```

        m_ctlXnsWindow.PushMedia(hMediaData);
    }
}

```

See Also

[Start, XnsSdkDevice의 OnNewMedia 이벤트](#)

SetPlaySpeed

Description

재생 속도를 설정합니다.

이 함수에서 제한하는 재생 속도의 최소/최대 값은 없으며, 다만 장치에 따라 최대 값을 넘어 설 경우, 최대 배속으로 자동 지정됩니다. 전송 속도가 너무 느릴 경우, 버퍼링(buffering)이 일어날 수 있습니다. 이 함수는 Playback 모드로 스트림을 오픈했을 때만 유효합니다. 즉, XnsSdkDevice::OpenStream()의 *nType*을 XMEDIA_PLAYBACK로 지정했을 때만 사용 가능합니다.

Syntax

```
void SetPlaySpeed(double dPlaySpeed);
```

Parameters

dPlaySpeed

[in] 재생 속도.

Example

[C++]

```
m_ctlXnsWindow.SetPlaySpeed(4.0);
```

See Also

[XnsSdkDevice::OpenStream, GetPlaySpeed](#)

GetPlaySpeed

Description

현재 설정된 재생 속도를 가져옵니다.

이 함수는 Playback 모드로 스트림을 오픈했을 때만 유효합니다. 즉, XnsSdkDevice::OpenStream()의 *nType*을 XMEDIA_PLAYBACK로 지정했을 때만 사용 가능합니다.

Syntax

```
double GetPlaySpeed(void);
```

Return Value

현재의 재생 속도.

Example

[C++]

```
double PlaySpeed = m_ctlXnsWindow.GetPlaySpeed();
```

See Also

[SetPlaySpeed](#), [XnsSdkDevice::OpenStream](#)

MoveFrameAdvance

Description

다음 프레임을 보여줍니다.

이 함수를 사용하려면, 재생 속도가 0으로 설정되어 있어야 하며, Playback 모드로 스트림을 오픈했을 때만 유효합니다. 즉, XnsSdkDevice::OpenStream()의 *nType*을 XMEDIA_PLAYBACK로 지정했을 때만 사용 가능합니다.

Syntax

```
void MoveFrameAdvance(void);
```

Example

[C++]

```
if (m_ctlXnsWindow.GetPlaySpeed() == 0)
    m_ctlXnsWindow.MoveFrameAdvance();
```

See Also

[SetPlaySpeed](#), [GetPlaySpeed](#), [XnsSdkDevice::OpenStream](#)

Control Buffer

ClearMediaBuffer

Description

버퍼의 내용을 삭제합니다.

재생 모드에서, 스트림을 수신하는 속도와 재생하는 속도의 차이 때문에 XnsSdkWindow는 스트림 데이터를 버퍼에 쌓아두게 됩니다. 만약, 버퍼에 데이터가 남아있는 상태에서 탐색(seek) 기능을 수행한다면, 몇몇 버퍼링된 스트림은 탐색 기능이 수행된 후, 화면에 표시됩니다. 이 함수는 탐색 기능을 사용하기 전에 버퍼링된 데이터를 삭제하기 위해 호출합니다.

Syntax

```
void ClearMediaBuffer(void);
```

Example

[C++]

```
m_ctlXnsWindow.ClearMediaBuffer();
long tPos = m_ctlXnsDevice.DateToTimet(2010, 9, 06, 15, 0, 0);
m_ctlXnsDevice.Seek(m_hDevice, 2, tPos);
```

See Also

[XnsSdkDevice::Seek](#), [DateToTimet](#)

IsBufferFull

Description

버퍼에 데이터가 가득 찼는지 확인합니다.

Syntax

```
BOOL IsBufferFull(void);
```

Return Value

버퍼가 가득 차 있으면 TRUE를 리턴합니다.

Example

[C++]

```
BOOL IsBufferFull = m_ctlXnsWindow.IsBufferFull();
if (IsBufferFull == TRUE)
```

```
m_ctlXnsWindow.ClearBuffer();
```

See Also

[IsBufferEmpty](#), [SetBufferSize](#), [GetBufferSize](#)

IsBufferEmpty

Description

버퍼가 비어있는지 체크합니다.

Syntax

```
BOOL IsBufferEmpty(void);
```

Return Value

버퍼가 비어 있으면 TRUE를 리턴합니다.

Example

[C++]

```
if(m_ctlXnsWindow.IsBufferEmpty())
    return;
```

See Also

[IsBufferFull](#), [SetBufferSize](#), [GetBufferSize](#)

SetBufferSize

Description

비디오 데이터를 저장할 버퍼 크기를 변경합니다. 이 함수가 호출되기 전까지 버퍼는 기본적으로 5 Mbytes가 할당됩니다. (이때, *nMinRemainingSize*는 0입니다.) 만약, 현재 남아있는 버퍼의 개수(즉, 영상 데이터 개수)가 *nMinRemainingSize*보다 작으면, OnPreBufferEmpty 이벤트가 발생합니다.

Syntax

```
void SetBufferSize(
    long nBufferSize,
    long nMinRemainingSize
```

);

Parameters

nBufferSize

[in] 버퍼 크기.

nMinRemainingSize

[in] OnPreBufferEmpty 이벤트를 발생시킬 최소 버퍼 개수. 즉, 버퍼에 있는 영상 데이터 개수가 이 값보다 작아질 때, OnPreBufferEmpty 이벤트가 발생함.

Example

[C++]

```
m_ctlXnsWindow.SetBufferSize(10*1024*1025, 5);
```

See Also

[GetBufferSize](#), [GetRemainingBufferSize](#), [OnPreBufferEmpty](#)

GetBufferSize

Description

현재 설정된 버퍼 크기를 반환합니다. 기본적으로 할당되는 크기는 5Mbytes(5x1024x1024)입니다.

Syntax

```
long GetBufferSize(void);
```

Return Value

설정된 버퍼 크기. (단위: 바이트)

Example

[C++]

```
long nBufferSize = m_ctlXnsWindow.GetBufferSize();
```

See Also

[SetBufferSize](#), [GetRemainingBufferSize](#)

GetRemainingBufferSize

Description

PreBufferEmpty 이벤트가 발생하는 최소 버퍼 개수를 반환합니다. (즉, SetBufferSize()의 *nMinRemainingSize* 파라미터 값을 의미합니다.)

Syntax

```
long GetRemainingBufferSize(void);
```

Return Value

PreBufferEmpty 이벤트가 발생하는 최소 버퍼 개수(영상 데이터 개수).

Example

[C++]

```
long RemainBuffer = m_ctlXnsWindow.GetRemainingBufferSize();
DisplayLong("GetRemainingBufferSize",RemainBuffer);
```

See Also

[SetBufferSize](#), [GetBufferSize](#)

Get Display Information

GetCurrentFrameTime

Description

현재 화면에 출력되고 있는 프레임의 시간을 가져옵니다.

Syntax

```
long GetCurrentFrameTime(void);
```

Return Value

현재 프레임의 시간을 time_t 포맷으로 반환합니다.

Example

[C++]

```
long tTime = m_ctlXnsWindow.GetCurrentFrameTime();

long Year, Mon, Day, Hour, Min, Sec;
CString str;

m_ctlXnsWindow.TimetToDate(tTime, &Year, &Mon, &Day, &Hour, &Min, &Sec);
str.Format("OnGetCurrentFrametime XnsWindow1 [%04d-%02d-%02d %02d:%02d:%02d\n",
Year, Mon, Day, Hour, Min, Sec);
OutputDebugString(str);
```

See Also

[OnFrameTimeChanged](#), [GetFirstFrameTime](#)

GetFirstFrameTime

Description

첫 번째 프레임의 시간을 가져옵니다.

Syntax

```
long GetFirstFrameTime(void);
```

Return Value

첫 번째 프레임의 시간을 time_t 포맷으로 반환합니다.

Example

[C++]

```
long tTime = m_ctlXnsWindow.GetFirstFrameTime();

long Year, Mon, Day, Hour, Min, Sec;
CString str;

m_ctlXnsWindow.TimetToDate(tTime, &Year, &Mon, &Day, &Hour, &Min, &Sec);
str.Format("GetFirstFrameTime XnsWindow1 [%04d-%02d-%02d %02d:%02d:%02d\n",
Year, Mon, Day, Hour, Min, Sec);
OutputDebugString(str);
```

See Also[GetCurrentFrameTime](#)

GetInputType – 사용하지 않음

Description

현재 XnsSdkWindow에서 재생되고 있는 미디어의 타입을 반환합니다.

이 함수를 사용하기 위해서는 애플리케이션이 OnNewMedia 이벤트의 이벤트 핸들러에서 얻은 hMediaData 값을 유지하고 있어야 합니다.

Syntax

```
long GetInputType(long hMediaData);
```

Parameters

hMediaData

[in] 미디어 데이터.

Return Value

미디어 타입.

- XINPUT_VIDEO(1): 비디오 데이터
- XINPUT_AUDIO(2): 오디오 데이터

Example

```
[C++]
```

```
long nInputType = m_ctlXnsWindow.GetInputType(hMediaData);
```

See Also[XnsSdkDevice의 OnNewMedia 이벤트](#)

GetFrameType – 사용하지 않음

Description

현재 재생되고 있는 프레임의 타입을 가져옵니다.

이 함수를 사용하기 위해서는 애플리케이션이 OnNewMedia 이벤트의 이벤트 핸들러에서 얻은 hMediaData 값을 유지하고 있어야 합니다.

Syntax

```
long GetFrameType(long hMediaData);
```

Parameters

hMediaData

[in] 미디어 데이터.

Return Value

프레임 타입.

- XFRAME_IVOP(1): Intra Frame
- XFRAME_PVOP(2): P-Frame(Predictive frame)
- XFRAME_BVOP(3): B-Frame(Bi-directional predictive frame)
- XFRAME_END(4): 마지막 프레임. (더 이상의 프레임이 남아있지 않음.)

Example

[C++]

```
long nFrameType = m_ctlXnsWindow.GetFrameType(hMediaData);
```

See Also

[GetFrameTime](#), [GetFrameSize](#)

GetFrameTime – 사용하지 않음

Description

재생되고 있는 스트림의 시간을 반환합니다.

이 함수를 사용하기 위해서는 애플리케이션이 OnNewMedia 이벤트의 이벤트 핸들러에서 얻은 *hMediaData* 값을 유지하고 있어야 합니다.

Syntax

```
long GetFrameTime(long hMediaData);
```

Parameters

hMediaData

[in] 미디어 데이터.

Return Value

현재 재생되고 있는 스트림의 프레임 시간을 time_t 형식으로 반환합니다.

Example

[C++]

```
long tFrameTime = m_ctlXnsWindow.GetFrameTime(hMediaData);
```

See Also

[GetCurrentFrameTime](#)

GetFrameSize – 사용하지 않음

Description

현재 재생되고 있는 프레임의 길이를 가져옵니다. 이 함수를 사용하기 위해서는 애플리케이션 이 OnNewMedia 이벤트의 이벤트 핸들러에서 얻은 hMediaData 값을 유지하고 있어야 합니다.

Syntax

```
long GetFrameSize(long hMediaData);
```

Parameters

hMediaData

[in] 미디어 데이터.

Return Value

프레임 데이터의 길이를 바이트 단위로 반환합니다.

Example

[C++]

```
long nFrameSize = m_ctlXnsWindow.GetFrameSize(hMediaData);
```

See Also

[GetFrameTime](#)

GetTimeStamp - 사용하지 않음

Description

미디어 데이터(*hMediaData*)의 타임 스탬프(timestamp) 값을 가져옵니다.

이 함수를 사용하기 위해서는 애플리케이션이 OnNewMedia 이벤트의 이벤트 핸들러에서 얻은 *hMediaData* 값을 유지하고 있어야 합니다.

타임 스탬프는 장비가 부팅된 후로 지금까지의 시간을 나타내는 값이며, millisecond 단위로 표시됩니다. 따라서, 애플리케이션은 이 값을 이용하여 현재 프레임과 이전 프레임의 시간 차를 구할 수 있습니다.

Syntax

```
long GetTimeStamp(long hMediaData);
```

Parameters

hMediaData

[in] 미디어 데이터.

Return Value

타임 스탬프.

Example

```
[C++]
```

```
long tFrameTime = m_ctlXnsWindow.GetFrameTimeStamp(hMediaData);
```

See Also

[GetFrameTime](#), [GetCurrentFrameTime](#), [GetFirstFrameTime](#)

SetBrightness

Description

이미지의 밝기 값을 설정합니다.

Syntax

```
void SetBrightness(long nValue);
```

Parameters

nValue

[in] 밝기 값. (0~100 사이의 값을 가짐.)

Example

[C++]

```
m_ctlXnsWindow.SetBrightness(20);
```

See Also

[GetBrightness](#), [SetContrast](#), [GetContrast](#)

GetBrightness

Description

설정된 밝기 값을 가져옵니다.

Syntax

```
long GetBrightness(void);
```

Return Value

현재의 밝기.

Example

[C++]

```
long Brightness = m_ctlXnsWindow.GetBrightness();
DisplayLong("GetBrightness", Brightness);
```

See Also

[SetBrightness](#), [SetContrast](#), [GetContrast](#)

SetContrast

Description

대비(contrast) 값을 설정합니다.

Syntax

```
void SetContrast(long nValue);
```

Parameters

nValue

[in] 대비 값. (1~100 사이의 값을 가짐.)

Example

[C++]

```
m_ctlXnsWindow.SetContrast(20);
```

See Also

[GetContrast](#), [SetBrightness](#), [GetBrightness](#)

GetContrast

Description

현재 설정된 대비 값을 가져옵니다.

Syntax

```
long GetContrast(void);
```

Return Value

설정된 대비 값.

Example

[C++]

```
long Contrast = m_ctlXnsWindow.GetContrast();
DisplayLong("GetContrast", Contrast);
```

See Also

[SetContrast](#), [SetBrightness](#), [GetBrightness](#)

SetDeinterlaceMode

Description

De-interlace 모드를 켜거나 끕니다.

Syntax

```
void SetDeinterlaceMode(BOOL bMode);
```

Parameters

bMode

[in] De-interlace 모드

1: De-interlace 모드를 켭니다.

0: De-interlace 모드를 끕니다.

Example

[C++]

```
m_ctlXnsWindow.SetDeinterlaceMode(1);
```

SetDefogMode

Description

Defog 모드를 켜거나 끕니다. 한 채널에 대해서만 지원합니다. 이미 다른 채널이 사용 중일 경우 사용이 불가능합니다.

Syntax

```
void SetDefogMode(BOOL bOn);
```

Parameters

bMode

[in] Defog 모드

- 1: Defog 모드를 켭니다.
- 0: Defog 모드를 끕니다.

Example

[C++]

```
m_ctlXnsWindow.SetDefogMode(1);
```

SetImageFlip

Description

영상 반전 모드를 켜거나 끕니다.

Syntax

```
long SetImageFlip(BOOL bHFlip, BOOL bVFlip);
```

Parameters

bHFlip

[in] 좌우 반전 모드

- 1 : 좌우 반전 모드를 켕니다.
- 0 : 좌우 반전 모드를 끕니다.

bVFlip

[in] 상하 반전 모드

- 1 : 상하 반전 모드를 켕니다.
- 0 : 상하 반전 모드를 끕니다.

Example

[C++]

```
m_ctlXnsWindow.SetImageFlip(1, 1);
```

GetVideoPixelFormat

Description

현재 화면에 출력되고 있는 디코딩 비디오의 Pixel format 정보를 가져옵니다.

Syntax

```
long GetVideoPixelFormat(void);
```

Return Value

Video Pixel format.

- XVIDEO_PIXEL_UNKNOWN (0x0000) : Unknown
- XVIDEO_PIXEL_R555 (0x0002) : 16Bit, R555
- XVIDEO_PIXEL_R565 (0x0003) : 16Bit, R565
- XVIDEO_PIXEL_BGR24 (0x0004) : 24Bit, RGB

- XVIDEO_PIXEL_BGR32 (0x0005) : 32Bit, Non Alpha Channel
- XVIDEO_PIXEL_BGRA (0x0006) : 32Bit, With Alpha Channel
- XVIDEO_PIXEL_YV12 (0x0101) : 12Bit, YV12

Example

[C++]

```
long pixel_format = m_ctlXnsWindow.GetVideoPixelFormat();
```

See Also

[OnVideoRawDataReceived](#)

SetAspectRatio

Description

종횡비 유지 기능을 켜거나 끕니다..

Syntax

```
void SetAspectRatio(BOOL bOn);
```

Parameters

bMode

[in] 종횡비 유지 기능

- 1: 종횡비를 유지합니다.
- 0: 종횡비를 유지하지 않습니다.

Example

[C++]

```
m_ctlXnsWindow.SetAspectRatio(1);
```

Miscellaneous

DisplayNoData

Description

녹화된 영상이 없는 경우, "No Data" 문구를 화면에 출력합니다.

이 함수는 미디어 탑입이 재생 모드일 때만 올바르게 동작합니다.

Syntax

```
void DisplayNoData(BOOL bState);
```

Parameters

bState

[in] "No Data" 문구 출력을 지정하는 플래그.

- 1: "No Data" 문구를 화면에 출력함.
- 0: "No Data" 문구를 화면에 출력하지 않음.

Example

[C++]

```
m_ctlXnsWindow.DisplayNoData(1);
```

SetCursor

Description

커서 탑입을 설정합니다.

Syntax

```
void SetCursor(long nCursorID);
```

Parameters

nCursorID

[in] 커서 탑입.

- 0: 기본 탑입.
- 1: 확대(zoom-in) 커서
- 2: 축소(zoom-out) 커서
- 3: 핸드(hand) 커서

- 4: 그랩(grab) 커서

Example

[C++]

```
m_ctlXnsWindow.setCursor(3);
```

GetLastError

Description

가장 최근에 발생한 에러 코드를 반환합니다.

Syntax

```
long GetLastError(void);
```

Return Value

최근에 발생한 에러 코드.

Example

[C++]

```
long LastError = m_ctlXnsWindow.GetLastError();
DisplayLong("ImageWidth", LastError);
```

DateToTimet

Description

날짜(년/월/일)와 시간(시/분/초)을 time_t 포맷으로 변환합니다.

Syntax

```
long DateToTimet(
    long nYear,
    long nMonth,
    long nDay,
    long nHour,
    long nMinute,
    long nSecond
```

```
);
```

Parameters

nYear

[in] 년(year)

nMonth

[in] 월(month)

nDay

[in] 일(day)

nHour

[in] 시(hour)

nMinute

[in] 분(minute)

nSecond

[in] 초(second)

Return Value

time_t 포맷으로 변환한 값.

Example

[C++]

```
long tStart = m_ctlXnsWindow.DateToTimet(2010, 8, 31, 0, 0, 10);
```

See Also

[TimetToDate](#)

TimetToDate

Description

time_t 포맷의 데이터를 날짜(년/월/일)와 시간(시/분/초)으로 변환합니다.

Syntax

```
void TimetToDate(
    long tTime,
    long* pnYear,
    long* pnMonth,
```

```

    long* pnDay,
    long* pnHour,
    long* pnMinute,
    long* pnSecond
);

```

Parameters

tTime

[in] 변환할 time_t 포맷의 데이터.

pnYear

[out] 년(year)

pnMonth

[out] 월(month)

pnDay

[out] 일(day)

pnHour

[out] 시(hour)

pnMinute

[out] 분(minute)

pnSecond

[out] 초(second)

Example

[C++]

```

long tTime = m_ctlXnsWindow.GetCurrentFrameTime();

long Year, Mon, Day, Hour, Min, Sec;
CString str;

m_ctlXnsWindow.TimetToDate(tTime, &Year, &Mon, &Day, &Hour, &Min, &Sec);

str.Format("OnGetCurrentFrametime XnsWindow1 [%04d-%02d-%02d %02d:%02d:%02d\n",
Year, Mon, Day, Hour, Min, Sec);
OutputDebugString(str);

```

See Also

[DateToTimet](#)

Events

이 절에서는 XnsSdkWindow에서 발생할 수 있는 이벤트 또는 XnsSdkWindow에 정의된 함수를 호출했을 때 발생하는 이벤트에 대해 설명합니다. 아래 나열된 함수들은 해당 이벤트가 발생했을 때, 호출되는 이벤트 핸들러입니다.

OnFrameTimeChanged

Description

화면에 표시되고 있는 프레임의 시간이 변경되면 발생하는 이벤트입니다. 즉, 영상 재생 시, 매 초마다 발생합니다.

Syntax

```
void OnFrameTimeChanged (long tFrameTime);
```

Parameters

tFrameTime

[in] 변경된 프레임 시간. 4 바이트의 time_t 포맷임.

Example

[C++]

```
void CTestSampleDlg::OnFrameTimeChangedXnssdkwindowctrl1(long tFrameTime)
{
    CString str;
    long Year, Mon, Day, Hour, Min, Sec;
    m_ctlXnsWindow.TimetToDate(tFrameTime, &Year, &Mon, &Day, &Hour, &Min, &Sec);
    str.Format("On FrameTime Changed XnsWindow1 [%04d-%02d-
02d %02d:%02d:%02d]\n", Year, Mon, Day, Hour, Min, Sec);
    OutputDebugString(str);
}
```

See Also

[GetCurrentFrameTime](#)

OnImageResolutionChanged

Description

원본 이미지의 해상도(resolution)가 변경되면 발생하는 이벤트입니다.

Syntax

```
void OnImageResolutionChanged (
    long nWidth,
    long nHeight
);
```

Parameters

nWidth

[in] 변경된 이미지의 너비.

nHeight

[in] 변경된 이미지의 높이.

Example

[C++]

```
void CTestSampleDlg::OnImageResolutionChangedXnssdkwindowctrl1(long nWidth, long
nHeight)
{
    CString str;
    str.Format("On ImageResolutionChanged XnsWindow1 nWidth[%d]
nHeight[%d]\n",nWidth, nHeight);
    OutputDebugString(str);
}
```

See Also

[GetImageWidth](#), [GetImageHeight](#)

OnMediaOn

Description

XnsSdkWindow 컴포넌트가 미디어 스트림을 수신했을 때 발생합니다.

Syntax

```
void OnMediaOn(void);
```

Example

[C++]

```
void CTestSampleDlg::OnMediaOnXnssdkwindowctrl1()
{
    OutputDebugString("On Media On Event XnsWindow1\n");
}
```

See Also

[XnsSdkDevice::OpenMedia](#)

OnMediaOff

Description

미디어 스트림의 수신이 중단되거나 완료되었을 때 발생합니다.

Syntax

```
void OnMediaOff(void);
```

Example

[C++]

```
void CTestSampleDlg::OnMediaOffXnssdkwindowctrl1()
{
    OutputDebugString("On Media Off Event XnsWindow1\n");
}
```

See Also

[OnMediaOn](#)

OnVideoRawDataReceived

Description

이 이벤트는 디코딩된 비디오 데이터가 생성되었을 때 발생합니다. 이 이벤트를 받기 위해서

[SetVideoRawDataReceived\(\)](#)를 먼저 호출해야 합니다. 이벤트를 취소하기 위해서 [UnsetVideoRawDataReceived\(\)](#) 를 호출할 수 있습니다.

Syntax

```
void OnVideoRawDataReceived
(
    long nDeviceID,
    long nControlID,
    long nWidth,
    long nHeight,
    long nVideoSize,
    long pVideoData
);
```

Parameters

nDeviceID

[in] 장비 ID.

nControlID

[in] Control ID.

nWidth

[in] 디코딩된 비디오데이터의 Width 해상도

nHeight

[in] 디코딩된 비디오데이터의 Height 해상도

nVideoSize

[in] 디코딩된 비디오데이터의 바이트크기

pVideoData

[in] 디코딩된 비디오데이터의 버퍼 주소값

Example

[C++]

```
void CVideoRawDataDlg::OnVideoRawDataReceivedXnssdkwindowctrl(long nDeviceID, long
nControlID, long nWidth, long nHeight, long nVideoSize, long pVideoData)
{
    m_nWidth = nWidth;
    m_nHeight = nHeight;
    unsigned char *pVideo = (unsigned char*)pVideoData;
```

```

ZeroMemory( &m_bmiHeader, sizeof(BITMAPINFOHEADER) );
m_bmiHeader.biWidth           = m_nWidth;
m_bmiHeader.biHeight          = m_nHeight*-1;
m_bmiHeader.biSize            = sizeof(BITMAPINFOHEADER);
m_bmiHeader.biPlanes          = 1;
m_bmiHeader.biBitCount        = 24;
m_bmiHeader.biCompression     = BI_RGB;
m_bmiHeader.biClrImportant    = 0;
m_bmiHeader.biSizeImage       = m_nWidth * m_nHeight * (m_bmiHeader.biBitCount / 8);

long pixel_format = m_ctrlXnsSdkWindow.GetVideoPixelFormat();
if (pixel_format == XVIDEO_PIXEL_BGRA)
{
    // 32bit RGB -> 24bit RGB
    for(int i = 0 ; i < nHeight*nWidth; i++)
    {
        memcpy(&m_pVideoBuf[i*3],&pVideo[i*4],3);
    }
}
else if (pixel_format == XVIDEO_PIXEL_R565)
{
    // 16bit RGB -> 24bit RGB
    unsigned short *pVideo16 = (unsigned short *)pVideo;
    int index = 0;
    for(int i = 0 ; i < nHeight*nWidth; i++)
    {
        unsigned short rgb16 = pVideo16[i];
        m_pVideoBuf[index++] = (rgb16 & 0x001f) << 3;
        m_pVideoBuf[index++] = (rgb16 & 0x07e0) >> 3;
        m_pVideoBuf[index++] = (rgb16 & 0xf800) >> 8;
    }
}

// Call OnPaint
Invalidate(FALSE);
}

```

See Also

[ControlID](#), [SetVideoRawDataReceived](#), [UnSetVideoRawDataReceived](#), [SaveSnapshotWithData](#),
[GetVideoPixelFormat](#)

OnLButtonDown

Description

원도우(XnSdkWindow)에서 마우스 왼쪽 버튼이 눌렸을 때 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 원도우 핸들과 부모 원도우의 원도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```
void OnLButtonDown(
    long nFlags,
    long nX,
    long nY
);
```

Parameters

nFlags

[in] WM_LBUTTONDOWN 원도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_RBUTTON: 마우스의 오른쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```
void CTestSampleDlg::OnLButtonDownXnssdkwindowctrl1(long nFlags, long nX, long nY)
{
    char str[100];
    sprintf(str, "On LButtonDown XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX,
nY);
```

```

        OutputDebugString(str);
    }
}

```

See Also

[OnLButtonUp](#), [OnRButtonDown](#), [OnRButtonUp](#)

OnLButtonUp

Description

윈도우(XnSdkWindow)에서 마우스의 왼쪽 버튼을 눌렀다가 떼는 경우에 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 윈도우 핸들과 부모 윈도우의 윈도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```

void OnLButtonUp(
    long nFlags,
    long nX,
    long nY
);

```

Parameters

nFlags

[in] WM_LBUTTONDOWN 윈도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_RBUTTON: 마우스의 오른쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```

void CTestSampleDlg::OnLButtonUpXnssdkwindowctrl1(long nFlags, long nX, long nY)
{
    char str[100];
}

```

```

sprintf(str, "On LButtonUp XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX, nY);
OutputDebugString(str);
}

```

See Also

[OnLButtonDown](#), [OnRButtonDown](#), [OnRButtonUp](#)

OnRButtonDown

Description

윈도우(XnsSdkWindow)에서 마우스 오른쪽 버튼이 눌렸을 때 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 원도우 핸들과 부모 원도우의 원도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```

void OnRButtonDown(
    long nFlags,
    long nX,
    long nY
);

```

Parameters

nFlags

[in] WM_RBUTTONDOWN 원도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_LBUTTON: 마우스의 왼쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```

void CTestSampleDlg::OnRButtonDownXnssdkwindowctrl1(long nFlags, long nX, long nY)
{

```

```

char str[100];
sprintf(str, "On RButtonDown XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX,
nY);
OutputDebugString(str);
}

```

See Also

[OnLButtonDown](#), [OnLButtonUp](#), [OnRButtonUp](#)

OnRButtonUp

Description

윈도우(XnSdkWindow)에서 마우스의 오른쪽 버튼을 눌렀다가 떼는 경우에 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 윈도우 핸들과 부모 윈도우의 윈도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```

void OnRButtonUp (
    long nFlags,
    long nX,
    long nY
);

```

Parameters

nFlags

[in] WM_RBUTTONDOWN 윈도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_LBUTTON: 마우스의 왼쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```
void CTestSampleDlg::OnRButtonUpXnssdkwindowctrl1(long nFlags, long nX, long nY)
{
    char str[100];
    sprintf(str, "On RButtonUp XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX, nY);
    OutputDebugString(str);
}
```

See Also

[OnLButtonDown](#), [OnLButtonUp](#), [OnRButtonDown](#)

OnLButtonDblClk

Description

윈도우(XnSdkWindow)에서 마우스 왼쪽 버튼을 더블클릭(double-click)했을 때 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 윈도우 핸들과 부모 윈도우의 윈도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```
void OnLButtonDblClk (
    long nFlags,
    long nX,
    long nY
);
```

Parameters

nFlags

[in] WM_LBUTTONDOWNDBLCLK 윈도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_RBUTTON: 마우스의 오른쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```
void CTestSampleDlg::OnLButtonDblClkXnsdkwindowctrl1(long nFlags, long nX, long nY)
{
    char str[100];
    sprintf(str, "On LButtonDblClk XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX,
nY);
    OutputDebugString(str);
}
```

See Also

[OnRButtonDblClk](#)

OnRButtonDblClk

Description

윈도우(XnSdkWindow)에서 마우스 오른쪽 버튼을 더블클릭(double-click)했을 때 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 윈도우 핸들과 부모 윈도우의 윈도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```
void OnRButtonDblClk (
    long nFlags,
    long nX,
    long nY
);
```

Parameters

nFlags

[in] WM_RBUTTONDOWNDBLCLK 윈도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_LBUTTON: 마우스의 왼쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```
void CTestSampleDlg::OnRButtonDblClkXnssdkwindowctrl1(long nFlags, long nX, long nY)
{
    char str[100];
    sprintf(str, "OnRButtonDblClk XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX,
nY);
    OutputDebugString(str);
}
```

See Also

[OnLButtonDblClk](#)

OnMouseMove

Description

마우스를 이동하면 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 원도우 핸들과 부모 원도우의 원도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```
void OnMouseMove (
    long nFlags,
    long nX,
    long nY
);
```

Parameters

nFlags

[in] WM_MOUSEMOVE 원도우 메시지에서 사용하는 플래그와 동일함.

- MK_CONTROL: [Ctrl] 키가 눌림.
- MK_MBUTTON: 마우스의 가운데 버튼이 눌림.
- MK_RBUTTON: 마우스의 오른쪽 버튼이 눌림.

nX

[in] 마우스 위치의 X좌표.

nY

[in] 마우스 위치의 Y좌표.

Example

[C++]

```
void CTestSampleDlg::OnMouseMoveXnssdkwindowctrl1(long nFlags, long nX, long nY)
{
    char str[100];
    sprintf(str, "On MouseMove XnsWindow1 nFlags[%d] nX[%d] nY[%d]\n", nFlags, nX, nY);
    OutputDebugString(str);
}
```

See Also

[OnMouseHover](#), [OnMouseLeave](#)

OnMouseHover

Description

마우스 포인터가 XnsSdkWindow로 진입하면 발생하는 이벤트입니다. 이 이벤트는 XnsSdkWindow의 윈도우 핸들과 부모 윈도우의 윈도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```
void OnMouseHover (void);
```

Example

[C++]

```
void CTestSampleDlg::OnMouseHoverXnssdkwindowctrl1()
{
    OutputDebugString("On MouseHover XnsWindow1\n");
}
```

See Also

[OnMouseMove](#), [OnMouseLeave](#)

OnMouseLeave

Description

마우스 포인터가 윈도우(XnsSdkWindow) 밖으로 이동했을 때 발생하는 이벤트입니다. 이 이벤

트는 XnsSdkWindow의 원도우 핸들과 부모 원도우의 원도우 핸들이 모두 NULL로 초기화 되었을 때만 유효합니다.

Syntax

```
void OnMouseLeave (void);
```

Example

[C++]

```
void CTestSampleDlg::OnMouseLeaveXnssdkwindowctrl1()
{
    OutputDebugString("On MouseLeave XnsWindow1\n");
}
```

See Also

[OnMouseMove](#), [OnMouseHover](#)

OnBufferFull

Description

이 이벤트는 XnsSdkWindow 컴포넌트의 버퍼가 가득 찼을 때 발생합니다. 애플리케이션은 ClearMediaBuffer()를 이용하여 버퍼를 비우거나 SetBufferSize()를 이용하여 버퍼의 크기를 조정할 수 있습니다.

Syntax

```
void OnBufferFull (void);
```

Example

[C++]

```
void CTestSampleDlg::OnBufferFullXnssdkwindowctrl1()
{
    m_ctlXnsWindow.ClearMediaBuffer();
}
```

See Also

[OnPreBufferFull](#), [ClearMediaBuffer](#), [SetBufferSize](#)

OnBufferEmpty

Description

이 이벤트는 XnsSdkWindow의 버퍼가 비었을 때 발생합니다.

Syntax

```
void OnBufferEmpty (void);
```

Example

[C++]

```
void CTestSampleDlg::OnBufferEmptyXnssdkwindowctrl1()
{
    OutputDebugString("On BufferEmpty XnsWindow1\n");
}
```

See Also

[OnPreBufferEmpty](#)

OnPreBufferFull

Description

이 이벤트는 XnsSdkWindow의 버퍼에 남아있는 용량이 버퍼 여유 공간의 최소값보다 작아지는 시점에 발생합니다. 애플리케이션은 SetBufferSize()를 이용하여 버퍼에서 허용하는 여유 공간의 최소값을 변경할 수 있습니다. (즉, SetBufferSize()의 *nMinRemainingSize* 파라미터 값을 조정하면 됩니다.) *nMinRemainingSize*에 기본적으로 설정되어 있는 값은 0입니다.

Syntax

```
void OnPreBufferFull (void);
```

Example

[C++]

```
void CTestSampleDlg::OnPreBufferFullXnssdkwindowctrl1()
{
    OutputDebugString("On PreBufferFull XnsWindow1\n");
}
```

See Also

[SetBufferSize](#), [OnBufferFull](#), [OnPreBufferEmpty](#)

OnPreBufferEmpty

Description

이 이벤트는 XnsSdkWindow의 버퍼에서 남아있는 데이터량이 지정된 값보다 작으면 발생합니다. 즉, 남아있는 데이터가 SetBufferSize()의 *nMinRemainingSize* 파라미터의 값보다 작으면 이 이벤트가 발생합니다. 기본적으로 *nMinRemainingSize*에 설정되어 있는 값은 0입니다.

Syntax

```
void OnPreBufferEmpty (void);
```

Example

[C++]

```
void CTestSampleDlg::OnPreBufferEmptyXnssdkwindowctrl1()
{
    OutputDebugString("On PreBufferEmpty XnsWindow1#\n");
}
```

See Also

[OnBufferEmpty](#), [OnPreBufferFull](#), [SetBufferSize](#)

OnEventStatusChanged

Description

이벤트의 상태가 변경되었을 때 발생하는 이벤트입니다. 이벤트 종류로는 영상 손실(video loss), 알람 발생, 움직임 감지(motion detection)가 있습니다.

Syntax

```
void OnEventStatusChanged (
    long nOldEventStatus,
    long nNewEventStatus
);
```

Parameters

nOldEventStatus

[in] 이전의 이벤트 상태.

nNewEventStatus

[in] 변경된 이벤트 상태.

이벤트 상태는 Bit-wise 방식으로 조합할 수 있으며, 다음과 같은 값을 가집니다.

표 3.2 Event Status

Event	Value
XEVENT_ALL	0x000000ff
XEVENT_SENSOR	0x00000001
XEVENT_MOTION	0x00000002
XEVENT_VIDEOLOSS	0x00000004

Example

[C++]

```
void CTestSampleDlg::OnEventStatusChangedXnssdkwindowctrl1(long nOldEventStatus, long
nNewEventStatus)
{
    CString str;
    str.Format("On EventStatusChanged XnsWindow1 nOldEventStatus[%d]
nNewEventStatus[%d]`n", nOldEventStatus, nNewEventStatus);
    OutputDebugString(str);
}
```

OnKeyDown

Description

이 이벤트는 non-시스템 키(non-system key)가 눌렸을 때 발생합니다.

Syntax

```
void OnKeyDown (
    long nChar,
    long nRepCnt,
    long nFlags
);
```

Parameters

nChar

[in] 키의 가상 키 코드(virtual key code).

nRepCnt

[in] 반복 횟수. (사용자가 키를 계속 눌러서 자동으로 반복된 키스트로크(keystroke)의 횟수임.)

nFlags

[in] 아래 표 3.3 참조.

표 3.3 OnKeyDown 이벤트의 nFlags가 나타내는 항목

Bit	Description
0~7	스캔 코드. (제조사에 따라 다름.)
8	기능키(function key)나 숫자패드의 키 같은 확장 키.
9~10	사용하지 않음.
11~12	MS 윈도우에서 내부적으로 사용함.
13	[Alt] 키가 함께 눌린 경우 1, 그렇지 않은 경우 0 임.
14	이전 키의 상태. 메시지를 보내기 전에 키가 눌려있으면 1, 눌렀다가 떼었으면 0 임.
15	키가 눌리면 0, 떼었으면 1 임.

Example

[C++]

```
void CTestSampleDlg::OnKeyDownXnssdkwindowctrl1(long nChar, long nRepCnt, long nFlags)
{
    CString str;
    str.Format("On KeyDown XnsWindow1 nChar[%d] nRepCnt[%d] nFlags[%d]\n", nChar,
    nRepCnt, nFlags);
    OutputDebugString(str);
}
```

See Also

[OnKeyUp](#)

OnKeyUp

Description

비시스템 키(non-system key)가 눌렸다가 떼어질 때 발생하는 이벤트입니다.

Syntax

```
void OnKeyUp (
    long nChar,
    long nRepCnt,
    long nFlags
);
```

Parameters

nChar

[in] 키의 가상 키 코드(virtual key code).

nRepCnt

[in] 반복 횟수. (사용자가 키를 계속 눌러서 자동으로 반복된 키스트로크(keystroke)의 횟수임.)

nFlags

[in] 아래 표 3.4 참조.

표 3.4 OnKeyUp 이벤트의 nFlags가 나타내는 항목

Bit	Description
0~7	스캔 코드. (제조사에 따라 다름.)
8	기능키(function key)나 숫자패드의 키 같은 확장 키.
9~10	사용하지 않음.
11~12	MS 윈도우에서 내부적으로 사용함.
13	[Alt] 키가 함께 눌린 경우 1, 그렇지 않은 경우 0 임.
14	이전 키의 상태. 메시지를 보내기 전에 키가 눌려있으면 1, 눌렸다가 떼었으면 0 임.
15	키가 눌리면 0, 떼었으면 1 임.

Example

[C++]

```
void CTestSampleDlg::OnKeyUpXnssdkwindowctrl1(long nChar, long nRepCnt, long nFlags)
{
    CString str;
    str.Format("On KeyDown XnsWindow1 nChar[%d] nRepCnt[%d] nFlags[%d]\n", nChar,
    nRepCnt, nFlags);
    OutputDebugString(str);
```

{}

See Also[OnKeyDown](#)

APPENDIX A

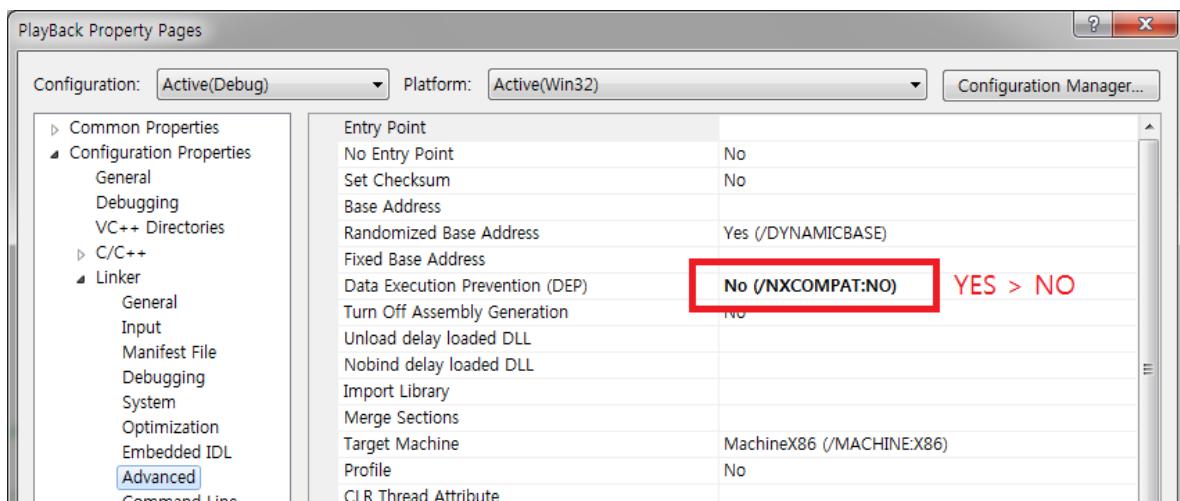
DEP 문제 해결 방법

DEP(데이터 실행 방지)는 프로그램을 모니터링하여 프로그램이 시스템 메모리를 안전하게 사용하게 함으로써 바이러스 및 다른 보안 위험으로부터 컴퓨터가 손상되는 것을 방지해 줄 수 있는 Windows의 보안 기능입니다. DEP 기능을 설정으로 인하여 SDK 프로그램이 올바르게 실행되지 않을 수 있습니다. 이 경우에는 다음과 같이 개발환경에서 직접 DEP를 해제할 수 있습니다.

C++ 프로젝트 설정

절차

- Step 1. [Solution Explorer]에서 프로젝트를 선택한 후, 마우스 오른쪽 버튼을 클릭하여 [Properties] 메뉴를 선택합니다. (37page 그림 참고.)
- Step 2. [Configuration Properties] > [Linker] > [Advanced] 메뉴를 선택합니다.
- Step 3. [Advanced] 메뉴에서 Data Execution Prevention(DEP) 항목이 YES로 되어 있다면, No로 변경합니다.



C# 프로젝트 설정

절차

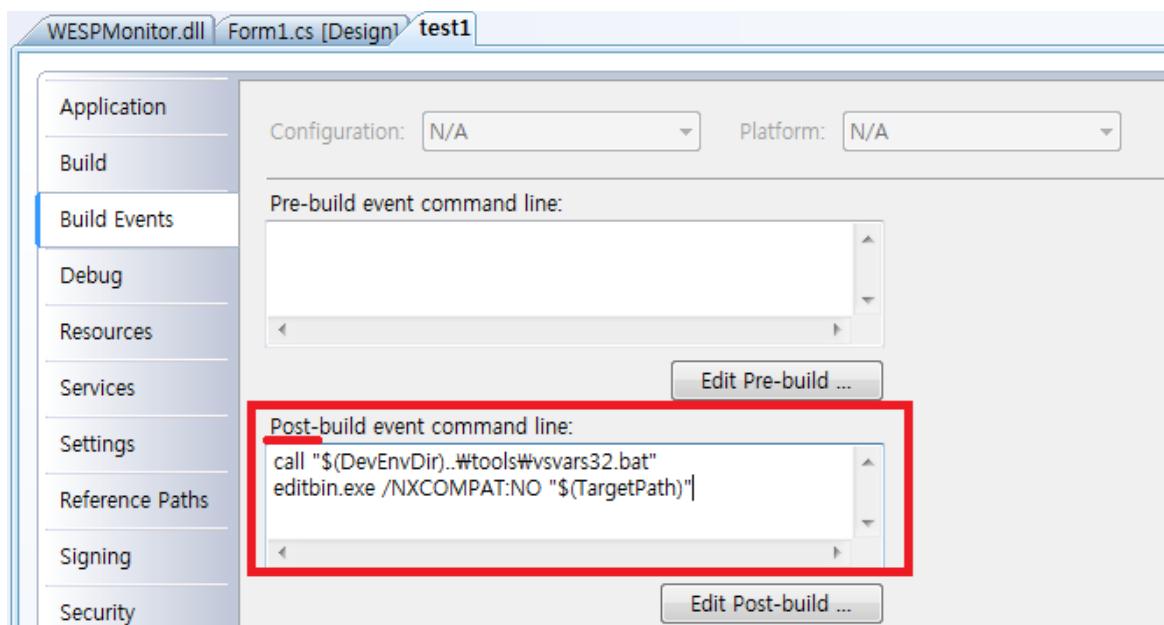
APPENDIX A

DEP 문제 해결 방법

Step 4. [Solution Explorer]에서 프로젝트를 선택한 후, 마우스 오른쪽 버튼을 클릭하여 [Properties] 메뉴를 선택합니다.

Step 5. [Build Events] 탭을 선택한 후, 아래와 같이 명령어를 입력합니다.

```
call "$(DevEnvDir)..#\tools\vsvars32.bat"  
editbin.exe /NXCOMPAT:NO "$(TargetPath)"
```



Visual Basic 프로젝트 설정

절차

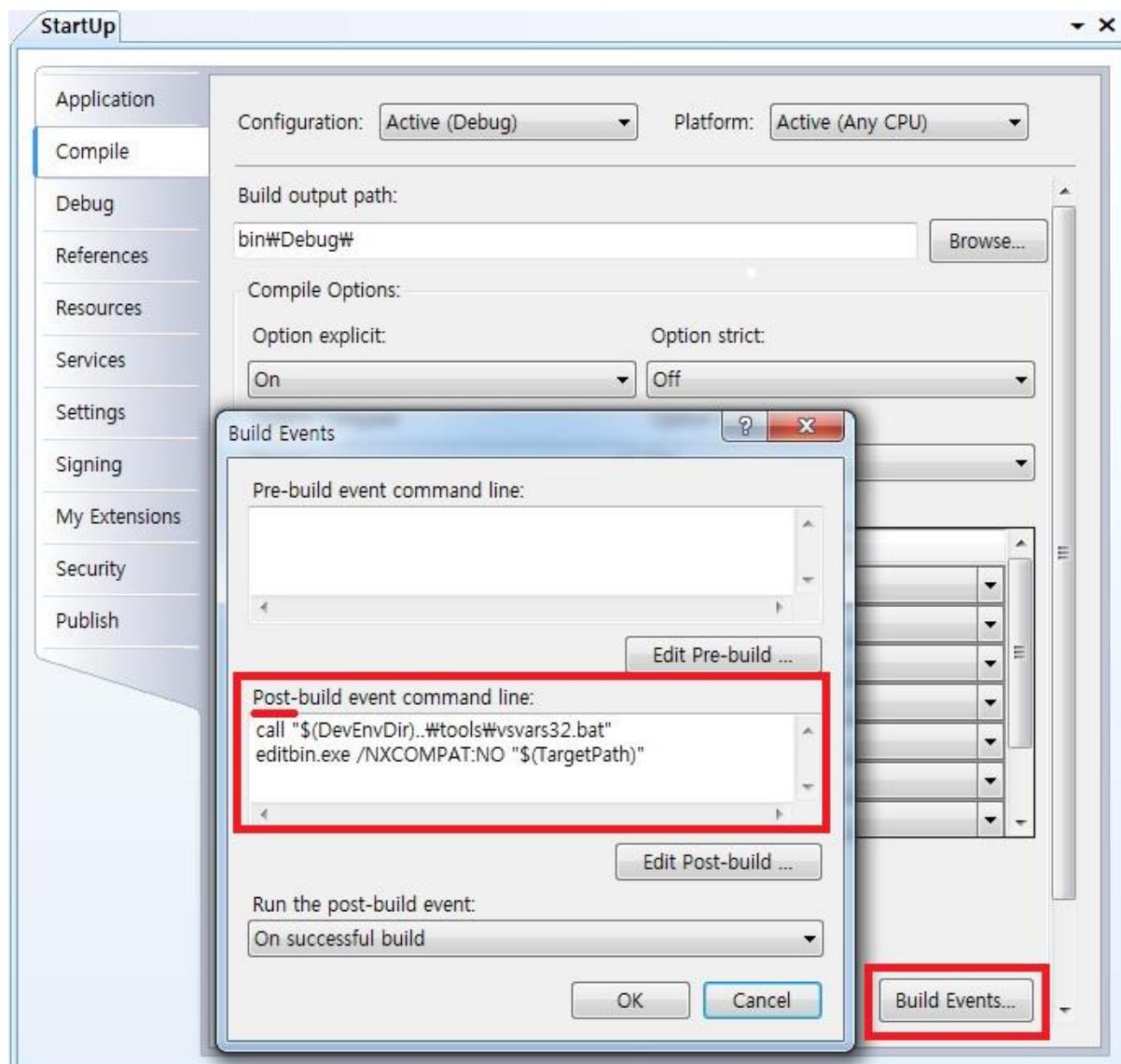
Step 6. [Solution Explorer]에서 프로젝트를 선택한 후, 마우스 오른쪽 버튼을 클릭하여 [Properties] 메뉴를 선택합니다.

Step 7. [Compile] 탭을 선택한 후, [Build Events] 버튼을 클릭하여, 아래와 같이 명령어를 입력합니다.

```
call "$(DevEnvDir)..#\tools\vsvars32.bat"  
editbin.exe /NXCOMPAT:NO "$(TargetPath)"
```

APPENDIX A

DEP 문제 해결 방법



APPENDIX B

에러 코드

이 장에서는 XNS ActiveX 라이브러리에서 반환하는 에러 코드에 대해 설명합니다.

Field	Value	Description
ERR_SUCCESS	0x00000000	함수 수행에 성공함.
ERR_UNKNOWN	0x00000001	알 수 없는 에러.
Initialize		
ERR_INVALID_CFG_FILE	0x00000010	잘못된 Config.xml 파일.
ERR_CFG_FILE_READ_ERROR	0x00000011	Config.xml 파일을 읽는데 실패함.
ERR_ALREADY_INITIALIZE	0x00000012	라이브러리가 이미 초기화 됨.
ERR_NOT_INITIALIZE	0x00000013	라이브러리가 초기화 되지 않음.
ERR_INVALID_DEVICE_HANDLE	0x00000014	잘못된 디바이스 핸들값
ERR_ALREADY_EXIST_DEVICEID	0x00000015	이미 존재하는 디바이스 아이디
Device Connection		
ERR_NOT_SUPPORTED_DEVICE	0x00000020	지원하지 않는 장비임.
ERR_EXCEED_CONNECTION	0x00000021	연결 가능한 수를 초과함.
ERR_NOT_CONNECT	0x00000022	장비가 연결되지 않음.
ERR_MAC_CONFLICT	0x00000023	MAC address 가 충돌이 남.
ERR_INVALID_VERSION	0x00000024	유효하지 않은 버전임.
ERR_INVALID_MODEL	0x00000025	유효하지 않은 모델임.
ERR_ALREADY_CONNECT	0x00000026	장비와 이미 연결되어 있음.
Device Disconnection (DVR/NVR)		
ERR_DISCONNECT_LOGOUT_BY_ADMIN	0x000000e0	다른 admin 계정 로그인으로 인해 접속해제됨.
ERR_DISCONNECT_TIME_CHANGE	0x000000e1	장비 시간 변경으로 접속해제됨.
ERR_DISCONNECT_HDD_ERASE	0x000000e2	HDD 데이터 삭제되어 접속해제됨

ERR_DISCONNECT_HDD_UNMOUNT	0x000000e3	HDD Unmount 되어 접속해제됨.
ERR_DISCONNECT_LOAD_FACTORY_DEFAULT	0x000000e4	장비 공장초기화로 인해 접속해제됨.
ERR_DISCONNECT_IMPORTED_CONFIG	0x000000e5	장비 설정 파일 Import 완료되어 접속해제됨.
ERR_DISCONNECT_SYSTEM_SHUTDOWN	0x000000e6	사용자가 장비의 Shutdown 키를 누르거나, 장비 메뉴를 통해 Shutdown 을 실행하여 접속해제됨.
ERR_DISCONNECT_REBOOT_MANUALLY	0x000000e7	SDK 의 재부팅 커맨드 실행으로 인해 접속해제됨.
ERR_DISCONNECT_REBOOT_UPGRADE	0x000000e8	F/W Upgrade 시작으로 접속해제됨.
ERR_DISCONNECT_NETWORK_CHANGE	0x000000e9	Network 변경으로 접속해제됨.

Login Error

ERR_INVALID_USERID	0x00000030	유효하지 않은 사용자 ID.
ERR_WRONG_PASSWORD	0x00000031	비밀번호가 맞지 않음.
ERR_LOGIN	0x00000032	로그인에 실패함.
ERR_ADMIN_CONFLICT	0x00000033	이미 다른 사용자가 관리자로 접속되어 있음.
ERR_USER_FULL	0x00000034	접속할 수 있는 최대 사용자 수를 초과함.

Query Request

ERR_XNS_BUSY	0x00000040	다른 작업 중임.
ERR_BAD_NETWORK	0x00000041	네트워크 연결 상태가 좋지 않음.
ERR_REQUEST_TIME_OUT	0x00000042	일정 시간 동안 사용자 요청에 대한 응답이 없음.
ERR_INVALID_PARAMETER	0x00000043	유효하지 않은 파라미터.
ERR_NO_PERMISSION	0x00000044	권한이 없음.
ERR_NOT_SUPPORTED_COMMAND	0x00000045	지원하지 않는 명령임.

File Transfer

ERR_INVALID_FILE	0x00000050	유효하지 않은 파일임.
ERR_UPLOAD_FAIL	0x00000051	파일 업로드에 실패함.
ERR_IMPORT_FAIL	0x00000052	파일 가져오기(import)에 실패함.
ERR_EXPORT_FAIL	0x00000053	파일 내보내기(export)에 실패함.

ERR_FILE_TRANSFERRING	0x00000054	파일 전송에 실패함.
Media Open		
ERR_MEDIA_ALREADY_OPENED	0x00000060	미디어가 이미 열려있음.
PTZ Control		
ERR_PTZ_CONFLICT	0x00000070	PTZ 작동 시, 충돌 발생.
XnsWindow Error		
ERR_INVALID_OBJID	0x00000080	유효하지 않은 Object ID.
ERR_ALREADY_RUN	0x00000081	이미 실행 중임.
ERR_MEDIA_NOT_OPENED	0x00000082	미디어가 열려있지 않음.
ERR_MEDIA_PAUSED	0x00000083	미디어가 일시 정지됨.
ERR_MEDIA_PLAYED	0x00000084	미디어가 재생 중임.
ERR_MEDIA_DISCONNECTED	0x00000085	미디어 연결이 끊어짐.
File Error		
ERR_FILE_ALREADY_OPENED	0x00000090	파일이 이미 열려있음.
ERR_FILE_NOT_OPENED	0x00000091	파일이 열려있지 않음.
ERR_FILE_OPEN	0x00000092	파일 오픈에 실패함.
ERR_FILE_SEEK	0x00000093	파일 탐색에 실패함.
ERR_FILE_READ	0x00000094	파일 읽기에 실패함.
ERR_FILE_WRITE	0x00000095	파일 쓰기에 실패함.
ERR_INVALID_FRAME	0x00000096	유효하지 않은 프레임.
ERR_WRITE_STOPPED	0x00000097	쓰기 작업이 중단됨.
ERR_WRITE_LIMITED	0x00000098	쓰기 작업이 제한됨.
ERR_WRITE_EMPTY	0x00000099	쓰기 작업이 진행된 데이터가 없음
ERR_DISK_FULL	0x0000009a	디스크에 여유 공간이 없음.
ERR_READ_END	0x0000009b	파일 읽기가 종료됨.
ERR_ALREADY_RECORDING	0x0000009c	이미 녹화됨.
ERR_BACKUP_DOING	0x0000009d	백업 중임.
기타		
ERR_INVALID_XML	0x000000b0	유효하지 않은 XML 파일.
ERR_OVERLAPPED	0x000000b1	OVERLAPPED IO 작업에 실패함
ERR_LIMITED	0x000000b2	사용이 제한됨

ERR_BUFFER_FULL	0x000000c0	버퍼에 여유 공간이 없음.
ERR_INVALID_SEQUENCE	0x000000c1	유효하지 않은 시퀀스(Sequence)임.
ERR_TALK_ALREADY_USE	0x000000c2	Talk 기능이 이미 사용중임.

APPENDIX C

UTC Time

이장에서는 UTC Time 에 대해 설명합니다.

UTC(Coordinated Universal Time) 협정시계시는 고정밀도 원자 시간 표준으로 컴퓨터 서버나 온라인 서비스 또는 여러 기관들은 UTC 시간을 통해 표준 시간대에 대해 중립적인 시간을 제공합니다. 전 세계의 표준 시간대는 아래와 같이 UTC에서의 양 또는 음의 오프셋으로 표시됩니다.

Time Zone	Locations
UTC−12:00	Baker Island, Howland Island (both uninhabited)
UTC−11:00	Samoa, American Samoa
UTC−10:00	Hawaii, Papeete
UTC−09:30	Marquesas Islands
UTC−09:00	Anchorage, Fairbanks, Juneau
UTC−08:00	Vancouver, Washington (state), Portland, Las Vegas, California, Baja California
UTC−07:00	Alberta, Colorado, Arizona, Chihuahua, Sonora
UTC−06:00	Chicago, Costa Rica, Dallas, El Salvador, Guatemala, Honduras, Houston, Manitoba, Mexico City, Nicaragua, Saskatchewan
UTC−05:00	Ottawa, Toronto, Montreal, Boston, New York, North Carolina, Washington D.C., Georgia, Miami, Cuba, Jamaica, Haiti, Panama, Colombia, Continental Ecuador, Peru
UTC−04:30	Venezuela
UTC−04:00	Nova Scotia, Dominican Republic, Puerto Rico, Trinidad and Tobago, Amazonas, Bolivia, Continental Chile, Paraguay, San Luis Province

<u>UTC -03:30</u>	<u>Newfoundland</u>
<u>UTC -03:00</u>	Rio de Janeiro, São Paulo, Argentina (except San Luis Province), Uruguay, Nuuk
<u>UTC -02:00</u>	Fernando de Noronha, South Georgia and the South Sandwich Islands
<u>UTC -01:00</u>	Azores, Cape Verde
<u>UTC</u>	Iceland, Faroe Islands, United Kingdom, Ireland, Continental Portugal, Madeira, Morocco, Senegal, Ghana, Côte d'Ivoire
<u>UTC+01:00</u>	Albania, Slovenia, Macedonia, Norway, Sweden, Denmark, Germany, the Netherlands, Belgium, Metropolitan France, Switzerland, Austria, Poland, Czech Republic, Slovakia, Hungary, Continental Spain, Italy, Croatia, Serbia, Kosovo, Bosnia and Herzegovina, Tunisia, Algeria, Nigeria, Cameroon, Angola, Kinshasa
<u>UTC+02:00</u>	Finland, Lithuania, Latvia, Estonia, Belarus, Ukraine, Romania, Bulgaria, Greece, Turkey, Cyprus, Syria, Lebanon, Jordan, Palestine, Israel, Egypt, Libya, Mozambique, Malawi, Zambia, Zimbabwe, South Africa
<u>UTC+03:00</u>	Samara, Iraq, Saudi Arabia, Yemen, Sudan, Ethiopia, Somalia, Kenya, Uganda, Tanzania, Madagascar
<u>UTC+03:30</u>	<u>Iran</u>
<u>UTC+04:00</u>	Georgia, Armenia, Azerbaijan, United Arab Emirates, Oman, Seychelles, Mauritius, Moscow, Saint Petersburg
<u>UTC+04:30</u>	<u>Afghanistan</u>
<u>UTC+05:00</u>	Sverdlovsk, Uzbekistan, Pakistan, Maldives, Kazakhstan
<u>UTC+05:30</u>	India, Sri Lanka
<u>UTC+05:45</u>	<u>Nepal</u>
<u>UTC+06:00</u>	Novosibirsk, Almaty, Bangladesh
<u>UTC+06:30</u>	Myanmar, Cocos Islands
<u>UTC+07:00</u>	Krasnoyarsk, Thailand, Vietnam, Jakarta
<u>UTC+08:00</u>	Irkutsk, Ulan Bator, China, Taiwan, Hong Kong, Philippines, Malaysia, Singapore, Western Australia
<u>UTC+09:00</u>	Zabaykalsky, Japan, North Korea, South Korea, East Timor
<u>UTC+09:30</u>	Northern Territory, South Australia
<u>UTC+10:00</u>	Victoria, Tasmania, Queensland, New South Wales, Primorsky
<u>UTC+10:30</u>	<u>Lord Howe Island</u>
<u>UTC+11:00</u>	Kamchatka, Solomon Islands, New Caledonia
<u>UTC+11:30</u>	<u>Norfolk Island</u>
<u>UTC+12:00</u>	Fiji, New Zealand
<u>UTC+12:45</u>	<u>Chatham Islands</u>

APPENDIX C

UTC Time

Events

UTC+13:00	Tonga
UTC+14:00	Line Islands

용어

N

Non-시스템 키

alt, ctrl 등의 특수키 조합을 사용하지 않은 일반 문자키를 말합니다.

약어

D

DLL

Dynamic Linking Library

DVR

Digital Video Recorder

M

MFC

Microsoft Foundation Class

O

OCX

Object Linking and Embedding (OLE) Custom Control

P

PTZ

Pan/Tilt/Zoom

S

SDK

Software Development Kit

색인

L

네트워크 장비	18
---------------	----

A

AcquireMediaAuthority	61
Application	18
AreaZoom	104
AutoScan	137

C

ChangeControlLocalName	60
ClearDraw	185
ClearMediaBuffer	195
ClearZoom	183
CloseFind	29
CloseMedia	66
CloseStream	68
Connect	43
ConnectNonBlock	44
control ID	48
Controllable module	47
ControlMenu	93
ControlPPtz	91
ControlPtz	86
ControlPtzFreeMove	88
ControlPtzSpeedZoomIn	89
ControlPtzSpeedZoomOut	90

CopyXml	131
CreateDevice	30
CreateFileReader	119

D

DateToTimet	133, 136, 209
DeleteConfigValue	131
DeleteDeviceProfile	128
DeletePreset	97
DeletePresetAll	98
Disconnect	46
DisplayNoData	207
DrawRect	184
DrawShape	185

E

ExecuteHPtz	94
ExportConfig	125

F

FindConfigChildeKeyList	132
FindDevice	25
FindModel	24
FindVendor	23
FrameAdvance	78

G

GetAddressType	36
GetBrightness	204
GetBufferSize	198
GetCameraCount	108
GetConfigPath	22
GetContrast	206
GetControlCapability	56
GetControlCount	49
GetControlDepth	51
GetControlID	177
GetControlLocalName	52
GetControlNumber	50
GetControlRemoteName	53
GetControlStatus	54
GetControlType	49
GetCurrentFrameTime	199
GetDeviceHandle	42
GetDeviceID	33, 176
GetDeviceStatus	34
GetErrorString	140
GetFileName	120
GetFindDouble	28
GetFindLong	26
GetFindSize	25
GetFindString	26
GetFirstFrameTime	200
GetFrameSize	203
GetFrameTime	202
GetFrameType	201
GetHttpPort	40
GetImageHeight	178
GetImageWidth	177
GetInputType	200
GetIP	37
GetLastError	209
GetLatestUpdateTime	41
GetMac	38

GetMediaSource	120
GetMediaType	179
GetModelName	35
GetParentControlID	60
GetPassword	41
GetPlaySpeed	194
GetPort	39
GetPreset	100
GetPresetCount	99
GetPresetList	95
GetPtzPos	102
GetRemainingBufferSize	198
GetTimeline	111
GetTimelineCount	109
GetTimelineHandle	121
GetTimeStamp	203
GetURL	38
GetUserID	40
GetVendorName	34
GetVersion	36
GetVolume	190
GetZoomInfo	182

H

HasAuthority	74
--------------------	----

I

ImportConfig	124
Initialize	21, 174
IPInstall	138
IsBufferEmpty	196
IsBufferFull	196
IsMedia	178
IsPlaying	73

L

LoadDeviceProfile 127

M

MoveFrameAdvance 195

O

OnAlarmOutControlFailed 168
OnBackupProgress 163
OnBufferEmpty 223
OnBufferFull 223
OnConnectFailed 144
OnControlCapabilityChanged 155
OnControlLocalNameChanged 141
OnControlRemoteNameChanged 169
OnControlStatusChaged 154
OnDeviceDetected 142
OnDeviceStatusChanged 152
OnEventStatusChanged 225
OnExportConfigProgress 151
OnFanBroken 167
OnFrameTimeChanged 212
OnGetPresetList 148
OnGetPtzPos 170
OnHDDFailed 161
OnHDDFulled 162
OnImageResolutionChanged 212
OnImportConfigProgress 150
OnIvEvent 159
OnKeyDown 226
OnKeyUp 227
OnLButtonDblClk 218
OnLButtonDown 214

OnLButtonUp 215
OnMediaOff 214
OnMediaOn 213
OnMotionEvent 157
OnMouseHover 222
OnMouseLeave 222
OnMouseMove 221
OnNewMedia 165
OnPasswordChanged 163
OnPreBufferEmpty 224
OnPreBufferFull 224
OnPtzControlFailed 167
OnRButtonDblClk 219
OnRButtonDown 216
OnRButtonUp 217
OnSearchCalendar 145
OnSearchDay 146
OnSensorEvent 158
OnUploadSoftwareProgress 149
OnVideoLossEvent 160
OpenMedia 63, 64
OpenStream 67

P

Pause 76
PauseReader 123
Play 75
PlayReader 122
PrintSnapshot 191
PushMedia 192

R

ReadConfigValue 129
ReleaseDevice 30

ReleaseFileReader	119
ReleaseMediaAuthority	62
ResumeAllStream	72
ResumeStream.....	71

S

SaveDeviceProfile	128
SaveSnapshot.....	191
SearchCalendar	79
SearchDay.....	80
Seek.....	77
SendAudioData	84
SetAlarm.....	85
SetBrightness	204
SetBufferSize	197
SetConnectionInfo.....	31
SetContrast	205
SetCursor	208
SetDateTimeFormat	188
SetDeinterlaceMode	206, 207
SetEventDiscardTime	140
SetListen	83
SetOSDOnOff.....	187
SetPlaySpeed	193
SetPreset	96
SetPtzPos	103
SetSound.....	189
SetTalk	82
SetVolume	189
Start.....	175
StartBackup.....	116
StartLocalRecording	113
Stop	176
StopBackup	118

StopLocalRecording.....	115
SuspendAllStream	70
SuspendStream	69

T

TimetToDate.....	134, 210
ToggleDvrRecording.....	115

U

UploadSoftware.....	123
---------------------	-----

W

WriteConfigValue.....	130
-----------------------	-----

X

XNS ActiveX.....	17
XnsSdkDevice.....	20

Z

Zoom1X	107
ZoomLevel.....	180
ZoomRate	181

OPEN SOURCE LICENSE REPORT ON THE PRODUCT

The software contains copyrighted software that is licensed under the GPL2.0, LGPL 2.1, MITLicense, zlib/libpng License, OpenSSL Combined License, Apache 2.0 License, bzip2 License, Common Public License.

You may obtain the Corresponding Source code from us for a period of three years after our last shipment of this product by sending email to help.cctv@hanwha.com. If you want to obtain the complete Corresponding Source code in the physical medium such as CD-ROM, the cost of physically performing source distribution might be charged.

- GPL 2.0 : gzip (<http://sourceforge.net/projects/gnuwin32>)
- LGPL 2.1 : Live555 (<http://www.live555.com/liveMedia>), FFmpeg (<http://ffmpeg.org>)
- MITLicense : Rapidxml - 1.13
- zlib/libpng Liscence: Nullsoft Scriptable Install System 2.46, tinyxml 2.6.2
- OpenSSL Combined License: OpenSSL
- Apache 2.0 License : Condor
- bzip2 License : Nullsoft Scriptable Install System 2.46
- Common Public License version 1.0 : Nullsoft Scriptable Install System 2.46 (<http://sourceforge.net/projects/nsis>)

Acknowledgement

This software includes modified version of Live555.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301,
USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can

get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights. We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that

redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files

and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above

on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the

Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is

restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT

HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:
Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

GNU LESSER GENERAL PUBLIC LICENSE

2.1

We used the FFmpeg in our application and we did not modify the source code. We modified the Live555 in the source code and used it in our application.

Copyright(C)

- Live555 (1991, 1999 Free Software Foundation, Inc.)
- FFmpeg(1991, 1999 Free Software Foundation, Inc.)

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301
USA Everyone is permitted to copy and distribute
verbatim copies of this license document, but changing it
is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away

your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that

a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the

freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs(which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and

appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the

Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License.

Also, you must do one of these things:

- a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the

modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any

other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our

decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.

Copyright (C) year name of author

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

The MIT License

Copyright (c) 2006, 2009 Marcin Kalicinski

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR

ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The zlib/libpng License

Copyright (c) 2011 Lee Thomason

Copyright (c) 1995-2012 Jean-loup Gailly and Mark Adler

Copyright (c) 1998-2012 Glenn Randers-Pehrson

Copyright (c) 2003-2005 Hector Mauricio Rodriguez Segura

Copyright (c) 1995-2015 Nullsoft and Contributors

Copyright (c) 2002-2008 Davide Pizzolato

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.;

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

OpenSSL License

Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of

this software must display the following acknowledgment:
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact opensslcore@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity

authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code

control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- a. You must give any other recipients of the Work or Derivative Works a copy of this License; and
- b. You must cause any modified files to carry prominent notices stating that You changed the files; and
- c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not

pertain to any part of the Derivative Works; and

d.If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licenser shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible

for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[] replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

bzip2 license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Julian Seward, Cambridge, UK.

jseward@acm.org

Common Public License version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and b) in the case of each subsequent Contributor:

i) changes to the Program, and

ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.

Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in,

the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions)

under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Special exception for LZMA compression module

Igor Pavlov and Amir Szekely, the authors of the LZMA compression module for NSIS, expressly permit you to statically or dynamically link your code (or bind by name) to the files from the LZMA compression module for NSIS without subjecting your linked code to the terms of the Common Public license version 1.0. Any modifications or additions to files from the LZMA compression module for NSIS, however, are subject to the terms of the Common Public License version 1.0.