

# Datastead TVideoGrabber SDK

## Table of contents

---

General .....	27
Online Help .....	27
Features .....	27
New features .....	29
Development platforms - System requirements .....	29
Support - Contact .....	29
Licensing .....	30
User guide .....	30
Datastead Encoder .....	30
Pre-requisties .....	32
Pre - requisities .....	32
Platform-dependent properties .....	33
Platform dependent properties .....	33
Device-dependent properties .....	33
Device dependent properties .....	33
Video sources .....	35
Video sources supported for preview and recording .....	35
USB and PCI video capture devices .....	36
Video capture devices .....	36
Preview .....	37
Video quality .....	38
Camera control .....	39
Detection of the video signal .....	40
Blackmagic Decklink cards .....	40
IP Cameras and other network streaming sources .....	41
The Datastead RTSP/RTMP/HTTP/ONVIF DirectShow source filter .....	41
ONVIF Cameras Discovery .....	42
ONVIF - Connecting to IP cameras through the ONVIF protocol .....	42
ONVIF - Retrieving camera information .....	43
ONVIF - PTZ .....	43
ONVIF Replay .....	44
ONVIF snapshot .....	45
Preview and recording of IP cameras and URL sources .....	45
Video stream of IP camera with audio of PC microphone or other audio capture device .....	46
IP cameras in MJPEG mode .....	47
IR Cut Filter of Axis cameras .....	53
Asynchronous vs Asynchronous connection .....	53
Auto-reconnection .....	55
NTP time of IP cameras .....	55
Using the TVideoGrabber http/mjpeg decoder when the Datastead RTSP filter is installed .....	55
Audio .....	56
Audio capture devices .....	56
Audio rendering .....	57
Audio levels and VU-meters .....	58
Recording .....	59

Recording methods and properties .....	59
Recording through the Datstead Encoder .....	63
Pause/resume during recording .....	65
Preroll recording (Backtimed recording) .....	66
Playing a video clip during recording .....	67
Dropped frames and audio/video sync problems .....	68
AES Encryption .....	69
AES Encryption Overview .....	69
Key .....	70
Realtime Encryption or Decryption .....	70
Batch Encryption or decryption .....	70
Disabling the encryption/decryption .....	71
Frame capture .....	71
Frame capture features .....	71
Graphic and text overlays .....	74
Setting and retrieving the overlay properties .....	74
Frame overlay vs window overlay .....	75
Overlays and aspect ratio .....	76
Image overlays .....	76
Text overlays .....	79
Graphic overlays .....	80
Chroma key .....	82
Overlays before or after transforms .....	82
To retrieve a pixel value .....	82
How to refresh the overlays while the video clip is paused .....	83
Mouse events .....	83
Player .....	84
Player features .....	84
Seamless playback .....	87
Current player state .....	87
Trackbar .....	88
Playing static images and animated GIFS .....	89
Playlist .....	90
Playing a clip from a TStream .....	93
StartPreview or StartRecording from video clips or URLs .....	93
AVI / ASF information and header attributes .....	94
Playing pictures .....	94
Synchronization of several player components .....	95
Video 360° .....	96
Decoding of 360° videos .....	96
PIP .....	97
PIP (Picture In Picture) .....	97
Streaming .....	98
Streaming through the Datstead Encoder .....	98
MMS streaming .....	99
Streaming using Newtek NDI .....	101
Video window - Display - Aspect Ratio - Dual display .....	102
Video window .....	102
Dual display .....	104
Multiple video windows .....	105

Transparency - Color keying .....	106
madVR video renderer .....	107
Third-party external renderers .....	107
Screen recording - desktop capture .....	108
screen capture and recording with or without cursor .....	108
screen capture and recording of a specified window .....	109
Recording only a part of the screen .....	110
Desktop capture .....	111
Recording on the fly of video clips and live streams .....	111
video clips and live streams recorded on the fly .....	111
Video clips built on the fly from Bitmap handles, BMP or JPEG files .....	112
Video clips built on the fly by passing bitmap handles, BMP or JPEG files .....	112
Video clips from a fixed set of BMP files or JPEG files merged into a clip .....	114
Video clip built from a fixed set of BMP files or JPEG files .....	114
Video processing .....	116
Zoom .....	116
Rotation, mirroring, vertical and horizontal flip .....	116
Cropping and zooming .....	117
Image adjustments (brightness, contrast, etc) .....	118
Deinterlacing .....	119
Motion detection .....	121
Motion detection .....	121
Motion ratio .....	122
Color intensity .....	122
Grid structure / grid sensitivity .....	123
Color / Greyscale .....	124
Video noise .....	125
Recording only when motion is detected .....	125
Mixing several video sources .....	126
Mixing several video sources into a single one .....	126
Reencoding .....	130
Reencoding of clips in batch mode .....	130
Reencoding of clips in preview or recording mode .....	133
Merging or splitting video clips .....	133
Video capture devices with multiplexed inputs .....	134
Video capture devices having multiplexed inputs .....	134
Opening a clip or an IP URL from a background thread without blocking .....	136
Opening a clip or an IP URL from a background thread without blocking the main thread .....	136
Synchronization of TVideoGrabber components .....	136
Synchronization of several TVideoGrabber components .....	136
Logo .....	137
Logo displayed in the video window .....	137
WPF .....	138
VidGrabWPF:TVideoGrabberWPF component .....	138
TVideoGrabber .....	138
Properties .....	138
AdjustOverlayAspectRatio .....	146
AdjustPixelAspectRatio .....	147
Aero .....	148



AnalogVideoStandard .....	149
AnalogVideoStandards .....	149
AnalogVideoStandardsCount .....	150
ApplicationPriority .....	150
ASFAudioBitRate .....	151
ASFAudioChannels .....	151
ASFBufferWindow .....	152
ASFDeinterlaceMode .....	152
ASFFixedFrameRate .....	153
ASFMediaServerPublishingPoint .....	154
ASFMediaServerRemovePublishingPointAfterDisconnect .....	154
ASFMediaServerTemplatePublishingPoint .....	155
ASFNetworkMaxUsers .....	156
ASFNetworkPort .....	156
ASFProfile .....	157
ASFProfileFromCustomFile .....	157
ASFProfiles .....	158
ASFProfilesCount .....	159
ASFProfileVersion .....	159
ASFVideoBitRate .....	160
ASFVideoFrameRate .....	160
ASFVideoHeight .....	161
ASFVideoMaxKeyFrameSpacing .....	162
ASFVideoQuality .....	162
ASFVideoWidth .....	163
AspectRatioToUse .....	163
AssociateAudioAndVideoDevices .....	164
AudioBalance .....	165
AudioChannelRenderMode .....	165
AudioCompressor .....	166
AudioCompressorName .....	167
AudioCompressors .....	167
AudioCompressorsCount .....	168
AudioDevice .....	168
AudioDeviceName .....	169
AudioDeviceRendering .....	170
AudioDevices .....	170
AudioDevicesCount .....	171
AudioFormat .....	171
AudioFormats .....	172
AudioInput .....	172
AudioInputBalance .....	173
AudioInputLevel .....	173
AudioInputMono .....	174
AudioInputs .....	174
AudioInputsCount .....	175
AudioPeakEvent .....	175
AudioRecording .....	175
AudioRenderer .....	176
AudioRendererName .....	177

AudioRenderers .....	177
AudioRenderersCount .....	178
AudioSource .....	178
AudioStreamNumber .....	179
AudioSyncAdjustment .....	179
AudioSyncAdjustmentEnabled .....	180
AudioVolume .....	180
AutoFileName .....	181
AutoFileNameDateTimeFormat .....	181
AutoFileNameMinDigits .....	182
AutoFilePrefix .....	183
AutoRefreshPreview .....	183
AutoStartPlayer .....	184
AVIDurationUpdated .....	184
AVIFormatOpenDML .....	185
AVIFormatOpenDMLCompatibilityIndex .....	186
BackgroundColor .....	186
BorderStyle .....	187
BurstCount .....	187
BurstInterval .....	188
BurstMode .....	188
BurstType .....	189
Busy .....	189
BusyCursor .....	190
CameraControlSettings .....	190
CaptureFileExt .....	190
ColorKey .....	191
ColorKeyEnabled .....	191
CompressionMode .....	192
CompressionType .....	192
Cropping_Enabled .....	193
Cropping_Height .....	193
Cropping_Outbounds .....	194
Cropping_Width .....	194
Cropping_X .....	195
Cropping_XMax .....	195
Cropping_Y .....	196
Cropping_YMax .....	196
Cropping_Zoom .....	196
CurrentFrameRate .....	197
CurrentState .....	197
DeliveredFrames .....	198
DirectShowFilters .....	198
DirectShowFiltersCount .....	198
Display_Active .....	199
Display_AlphaBlendEnabled .....	199
Display_AlphaBlendValue .....	200
Display_AspectRatio .....	201
Display_AutoSize .....	202
Display_Embedded .....	202

Display_FullScreen .....	203
Display_Height .....	204
Display_Left .....	204
Display_Monitor .....	205
Display_MouseMovesWindow .....	206
Display_PanScanRatio .....	206
Display_StayOnTop .....	207
Display_Top .....	208
Display_TransparentColorEnabled .....	208
Display_TransparentColorValue .....	209
Display_VideoHeight .....	210
Display_VideoPortEnabled .....	210
Display_VideoWidth .....	211
Display_VideoWindowHandle .....	212
Display_Visible .....	212
Display_Width .....	213
DroppedFrames .....	213
DroppedFramesPollingInterval .....	214
DualDisplay_Active .....	214
DualDisplay_AlphaBlendEnabled .....	215
DualDisplay_AlphaBlendValue .....	216
DualDisplay_AspectRatio .....	216
DualDisplay_AutoSize .....	217
DualDisplay_Embedded .....	217
DualDisplay_FullScreen .....	218
DualDisplay_Height .....	219
DualDisplay_Left .....	219
DualDisplay_Monitor .....	220
DualDisplay_MouseMovesWindow .....	221
DualDisplay_PanScanRatio .....	221
DualDisplay_StayOnTop .....	222
DualDisplay_Top .....	223
DualDisplay_TransparentColorEnabled .....	223
DualDisplay_TransparentColorValue .....	224
DualDisplay_VideoHeight .....	225
DualDisplay_VideoPortEnabled .....	225
DualDisplay_VideoWidth .....	226
DualDisplay_VideoWindowHandle .....	227
DualDisplay_Visible .....	227
DualDisplay_Width .....	228
DVDateTimeEnabled .....	229
DVDiscontinuityMinimumInterval .....	229
DVDTitle .....	230
DVEncoder_VideoFormat .....	230
DVEncoder_VideoResolution .....	231
DVEncoder_VideoStandard .....	231
DVRecordingInNativeFormatSeparatesStreams .....	232
DVReduceFrameRate .....	232
DVRgb219 .....	233
DVTimeCodeEnabled .....	233

EncryptionMethod .....	234
EventNotificationSynchronone .....	234
ExtraDLLPath .....	235
FixFlickerOrBlackCapture .....	236
FrameCaptureHeight .....	236
FrameCaptureWidth .....	237
FrameCaptureWithoutOverlay .....	237
FrameCaptureZoomSize .....	238
FrameGrabber .....	239
FrameGrabberCurrentRGBFormat .....	240
FrameGrabberRGBFormat .....	240
FrameNumberStartsFromZero .....	241
FrameRate .....	241
FramerateDivider .....	242
GetLastFrameWaitTimeoutMs .....	242
HoldRecording .....	243
ImageOverlay_AlphaBlend .....	244
ImageOverlay_AlphaBlendValue .....	244
ImageOverlay_ChromaKey .....	245
ImageOverlay_ChromaKeyLeewayPercent .....	246
ImageOverlay_ChromaKeyRGBColor .....	246
ImageOverlay_Height .....	247
ImageOverlay_LeftLocation .....	247
ImageOverlay_RotationAngle .....	248
ImageOverlay_StretchToVideoSize .....	248
ImageOverlay_TopLocation .....	249
ImageOverlay_Transparent .....	250
ImageOverlay_TransparentColorValue .....	251
ImageOverlay_UseTransparentColor .....	251
ImageOverlay_VideoAlignment .....	252
ImageOverlay_Width .....	253
ImageOverlayEnabled .....	253
ImageOverlaySelector .....	254
ImageRatio .....	256
InFrameProgressEvent .....	256
IPCameraURL .....	257
IsAnalogVideoDecoderAvailable .....	257
IsAudioCrossbarAvailable .....	258
IsAudioInputBalanceAvailable .....	258
IsCameraControlAvailable .....	259
IsDigitalVideoIn .....	259
IsDVCommandAvailable .....	260
IsHorizontalSyncLocked .....	260
IsMPEGStream .....	261
IsPlayerAudioStreamAvailable .....	261
IsPlayerVideoStreamAvailable .....	262
IsRecordingPaused .....	262
IsTimeCodeReaderAvailable .....	263
IsTVAudioAvailable .....	263
IsTVAutoTuneRunning .....	264

IsTVTunerAvailable .....	264
IsVideoControlAvailable .....	265
IsVideoCrossbarAvailable .....	265
IsVideoInterlaced .....	265
IsVideoPortAvailable .....	266
IsVideoQualityAvailable .....	267
IsWDMVideoDriver .....	267
JPEGPerformance .....	267
JPEGProgressiveDisplay .....	268
JPEGQuality .....	269
Last_BurstFrameCapture_FileName .....	269
Last_CaptureFrameTo_FileName .....	270
Last_Clip_Played .....	270
Last_Recording_FileName .....	271
LogoDisplayed .....	271
LogoLayout .....	272
MixAudioSamples_CurrentSourceLevel .....	272
MixAudioSamples_ExternalSourceLevel .....	273
Mixer_MosaicColumns .....	273
Mixer_MosaicLines .....	274
MotionDetector_CompareBlue .....	274
MotionDetector_CompareGreen .....	275
MotionDetector_CompareRed .....	275
MotionDetector_Enabled .....	276
MotionDetector_GlobalMotionRatio .....	277
MotionDetector_GreyScale .....	277
MotionDetector_Grid .....	278
MotionDetector_GridXCount .....	279
MotionDetector_GridYCount .....	279
MotionDetector_IsGridValid .....	280
MotionDetector_MaxDetectionsPerSecond .....	281
MotionDetector_ReduceCPULoad .....	281
MotionDetector_ReduceVideoNoise .....	282
MotionDetector_Triggered .....	283
MouseWheelEventEnabled .....	283
MpegStreamType .....	284
MultiplexedInputEmulation .....	285
MultiplexedRole .....	285
MultiplexedStabilizationDelay .....	286
MultiplexedSwitchDelay .....	286
MuteAudioRendering .....	287
NDIFormatType .....	287
NDIName .....	288
NetworkStreaming .....	288
NetworkStreamingType .....	289
NormalCursor .....	289
NotificationMethod .....	290
NotificationPriority .....	290
NotificationSleepTime .....	291
OnFrameBitmapEventSynchrone .....	291

OpenURLAsync .....	292
OverlayAfterTransform .....	292
ParentWindow .....	293
PlayerAudioCodec .....	294
PlayerAudioRendering .....	294
PlayerDuration .....	295
PlayerDVSize .....	296
PlayerFastSeekSpeedRatio .....	296
PlayerFileName .....	297
PlayerForcedCodec .....	298
PlayerFrameCount .....	299
PlayerFramePosition .....	300
PlayerFrameRate .....	300
PlayerHwAccel .....	301
PlayerOpenProgressPercent .....	301
PlayerRefreshPausedDisplay .....	301
PlayerRefreshPausedDisplayFrameRate .....	302
PlayerSpeedRatio .....	303
PlayerSpeedRatioConstantAudioPitch .....	303
PlayerState .....	304
PlayerTimePosition .....	304
PlayerTrackBar .....	305
PlayerTrackBarScale .....	306
PlayerTrackBarSynchronise .....	307
PlayerVideoCodec .....	307
PlaylistIndex .....	308
PreallocCapFileCopiedAfterRecording .....	308
PreallocCapFileEnabled .....	309
PreallocCapFileName .....	310
PreallocCapFileSizeInMB .....	310
PreviewZoomSize .....	311
RawAudioSampleCapture .....	312
RawCaptureAsyncEvent .....	312
RawSampleCaptureLocation .....	313
RawVideoSampleCapture .....	313
RecordingAudioBitRate .....	314
RecordingBacktimedFramesCount .....	314
RecordingCanPause .....	315
RecordingDuration .....	315
RecordingFileName .....	316
RecordingFileSizeMaxInMB .....	317
RecordingFourCC .....	317
RecordingHeight .....	318
RecordingInNativeFormat .....	319
RecordingMethod .....	320
RecordingOnMotion_Enabled .....	320
RecordingOnMotion_MotionThreshold .....	321
RecordingOnMotion_NoMotionPauseDelayMs .....	322
RecordingPauseCreatesNewFile .....	322
RecordingSize .....	323

RecordingTimer .....	324
RecordingTimerInterval .....	325
RecordingVideoBitRate .....	325
RecordingWidth .....	326
Reencoding_IncludeAudioStream .....	326
Reencoding_IncludeVideoStream .....	327
Reencoding_Method .....	327
Reencoding_NewVideoClip .....	328
Reencoding_SourceVideoClip .....	329
Reencoding_StartFrame .....	329
Reencoding_StartTime .....	330
Reencoding_StopFrame .....	330
Reencoding_StopTime .....	331
Reencoding_UseAudioCompressor .....	331
Reencoding_UseFrameGrabber .....	332
Reencoding_UseVideoCompressor .....	332
Reencoding_WMVOutput .....	333
ScreenRecordingLayeredWindows .....	333
ScreenRecordingMonitor .....	334
ScreenRecordingNonVisibleWindows .....	334
ScreenRecordingSizePercent .....	335
ScreenRecordingThroughClipboard .....	335
ScreenRecordingWithCursor .....	336
SendToDV_DeviceIndex .....	336
ShapeOverlay .....	337
ShapeOverlayEnabled .....	338
SourceStream .....	339
SpeakerBalance .....	339
SpeakerControl .....	340
SpeakerVolume .....	340
StoragePath .....	341
StoreDeviceSettingsInRegistry .....	341
StreamingURL .....	342
StreamInterface_Format .....	342
StreamInterface_FrameRate .....	343
StreamInterface_IsRealTime .....	343
SyncCommands .....	344
SynchronizationRole .....	344
Synchronized .....	345
SyncPreview .....	346
SystemTempPath .....	346
TextOverlay_Align .....	347
TextOverlay_AlphaBlend .....	348
TextOverlay_AlphaBlendValue .....	348
TextOverlay_BkColor .....	348
TextOverlay_Enabled .....	349
TextOverlay_Font .....	350
TextOverlay_FontColor .....	351
TextOverlay_Left .....	352
TextOverlay_Right .....	353

TextOverlay_Scrolling .....	354
TextOverlay_ScrollingSpeed .....	354
TextOverlay_Selector .....	355
TextOverlay_Shadow .....	358
TextOverlay_ShadowColor .....	358
TextOverlay_ShadowDirection .....	359
TextOverlay_String .....	360
TextOverlay_Top .....	362
TextOverlay_Transparent .....	363
TextOverlay_VideoAlignment .....	363
ThirdPartyDeinterlacer .....	364
TranslateMouseCoordinates .....	364
TunerFrequency .....	365
TunerMode .....	366
TVChannel .....	366
TVCountryCode .....	367
TVTunerInputType .....	367
TVUseFrequencyOverrides .....	368
UniqueID .....	368
UseClock .....	369
v360_AspectRatio .....	369
v360_Enabled .....	370
v360_MasterAngle .....	370
v360_MouseAction .....	371
v360_MouseActionPercent .....	371
VCRHorizontalLocking .....	372
Version .....	372
VideoCompression_DataRate .....	372
VideoCompression_KeyFrameRate .....	373
VideoCompression_PFramesPerKeyFrame .....	374
VideoCompression_Quality .....	374
VideoCompression_WindowSize .....	375
VideoCompressor .....	375
VideoCompressorName .....	376
VideoCompressors .....	377
VideoCompressorsCount .....	377
VideoControlSettings .....	378
VideoCursor .....	378
VideoDevice .....	379
VideoDeviceName .....	379
VideoDevices .....	380
VideoDevicesCount .....	380
VideoDevicesId .....	381
VideoDoubleBuffered .....	382
VideoFormat .....	382
VideoFormats .....	383
VideoFormatsCount .....	384
VideoFromImages_BitmapsSortedBy .....	384
VideoFromImages_RepeatIndefinitely .....	385
VideoFromImages_SourceDirectory .....	385



VideoFromImages_TemporaryFile .....	386
VideoHeight .....	386
VideoHeight_PreferedAspectRatio .....	387
VideoInput .....	387
VideoInputs .....	388
VideoInputsCount .....	389
VideoPlayableWhileRecording .....	389
VideoProcessing_Brightness .....	390
VideoProcessing_Contrast .....	390
VideoProcessing_Deinterlacing .....	391
VideoProcessing_FlipHorizontal .....	391
VideoProcessing_FlipVertical .....	392
VideoProcessing_GrayScale .....	392
VideoProcessing_Hue .....	393
VideoProcessing_InvertColors .....	393
VideoProcessing_Pixellization .....	394
VideoProcessing_Rotation .....	394
VideoProcessing_RotationCustomAngle .....	395
VideoProcessing_Saturation .....	396
VideoQualitySettings .....	396
VideoRenderer .....	397
VideoRendererExternal .....	397
VideoRendererExternalIndex .....	398
VideoRendererPriority .....	399
VideoSize .....	400
VideoSizes .....	400
VideoSizesCount .....	401
VideoSource .....	401
VideoSource_FileOrURL .....	402
VideoSource_FileOrURL_StartTime .....	403
VideoSource_FileOrURL_StopTime .....	404
VideoSources .....	404
VideoSourcesCount .....	405
VideoStreamNumber .....	405
VideoSubtype .....	405
VideoSubtypes .....	406
VideoSubtypesCount .....	406
VideoVisibleWhenStopped .....	407
VideoWidth .....	407
VideoWidth_PreferedAspectRatio .....	408
Visible .....	409
VUMeter .....	409
WebcamStillCaptureButton .....	410
ZoomCoeff .....	411
ZoomXCenter .....	411
ZoomYCenter .....	411
Methods .....	412
About .....	418
AnalogVideoStandardIndex .....	419
ASFStreaming_GetAuthorizationList .....	419

ASFStreaming_GetConnectedClients .....	420
ASFStreaming_GetConnectedClientsCount .....	420
ASFStreaming_ResetAuthorizations .....	421
ASFStreaming_SetAuthorization .....	421
AssociateMultiplexedSlave .....	422
AudioCompressorIndex .....	422
AudioDeviceIndex .....	423
AudioInputIndex .....	424
AudioRendererIndex .....	424
AVIDuration .....	425
AVIHeaderInfo .....	425
AVIInfo .....	426
AVIInfo2 .....	428
CameraControlAuto .....	429
CameraControlDefault .....	429
CameraControlMax .....	430
CameraControlMin .....	430
CameraControlStep .....	431
CameraControlValue .....	431
Cancel .....	432
CanProcessMessages .....	432
CaptureFrameRenderedTo .....	433
CaptureFrameSyncTo .....	433
CaptureFrameTo .....	434
ClearHeaderAttributes .....	434
ClosePlayer .....	435
Create .....	435
CreatePreallocCapFile .....	436
Decrypt_File .....	436
Destroy .....	436
Display_SetLocation .....	437
DrawBitmapOverFrame .....	437
DualDisplay_SetLocation .....	439
DVDInfo .....	439
EnableMultiplexedInput .....	440
EnableThreadMode .....	441
Encoder_CloseOutputFile .....	441
Encoder_GetInt .....	442
Encoder_NewOutputFile .....	442
Encoder_Pause .....	442
Encoder_Resume .....	443
Encoder_SetInt .....	443
Encoder_SetStr .....	444
Encoders_CreateInstanceForRecording .....	444
Encoders_CreateInstanceForStreaming .....	445
Encoders_RemoveAllInstances .....	445
Encoders_RemoveInstance .....	446
Encrypt_File .....	446
EnumerateWindows .....	446
Facebook_GoLive_ReturnStreamURL .....	447

FastForwardPlayer .....	448
FindIndexInListByName .....	449
GetAudioCodec .....	450
GetCameraExposure .....	450
GetCameraExposureAsString .....	450
GetDisplayActive .....	451
GetDisplayAlphaBlendEnabled .....	451
GetDisplayAlphaBlendValue .....	451
GetDisplayAspectRatio .....	452
GetDisplayAutoSize .....	452
GetDisplayEmbedded .....	452
GetDisplayFullScreen .....	453
GetDisplayHeight .....	453
GetDisplayLeft .....	453
GetDisplayMonitor .....	454
GetDisplayMouseMovesWindow .....	454
GetDisplayPanScanRatio .....	455
GetDisplayStayOnTop .....	455
GetDisplayTop .....	455
GetDisplayTransparentColorEnabled .....	456
GetDisplayTransparentColorValue .....	456
GetDisplayVideoHeight .....	456
GetDisplayVideoPortEnabled .....	457
GetDisplayVideoWidth .....	457
GetDisplayVideoWindowHandle .....	457
GetDisplayVisible .....	458
GetDisplayWidth .....	458
GetFrameInfo .....	458
GetFrameInfoString .....	459
GetFWCam1394 .....	460
GetFWCam1394List .....	461
GetImageOverlay_AlphaBlend .....	461
GetImageOverlay_AlphaBlendValue .....	461
GetImageOverlay_ChromaKey .....	462
GetImageOverlay_ChromaKeyLeewayPercent .....	462
GetImageOverlay_ChromaKeyRGBColor .....	462
GetImageOverlay_Enabled .....	463
GetImageOverlay_Height .....	463
GetImageOverlay_LeftLocation .....	463
GetImageOverlay_RotationAngle .....	464
GetImageOverlay_StretchToVideoSize .....	464
GetImageOverlay_TargetDisplay .....	464
GetImageOverlay_TopLocation .....	464
GetImageOverlay_Transparent .....	465
GetImageOverlay_TransparentColorValue .....	465
GetImageOverlay_UseTransparentColor .....	465
GetImageOverlay_Width .....	466
GetItemNameFromList .....	466
GetLastAverageStreamValue .....	466
GetLastErrorMessages .....	467

GetLastFrameAsHBITMAP .....	467
GetLastFrameAsTBitmap .....	469
GetLastFrameBitmapBits .....	469
GetLastFrameBitmapBits2 .....	470
GetLogString .....	471
GetMiscDeviceControl .....	471
GetNearestVideoHeight .....	472
GetNearestVideoSize .....	472
GetNearestVideoWidth .....	473
GetPixelsDistance .....	473
GetPlaylist .....	474
GetRGBPixelAt .....	474
GetTextOverlay_Align .....	476
GetTextOverlay_AlphaBlend .....	476
GetTextOverlay_AlphaBlendValue .....	476
GetTextOverlay_BkColor .....	477
GetTextOverlay_Enabled .....	477
GetTextOverlay_Font .....	477
GetTextOverlay_GradientColor .....	478
GetTextOverlay_GradientMode .....	478
GetTextOverlay_HighResFont .....	478
GetTextOverlay_Left .....	479
GetTextOverlay_Right .....	479
GetTextOverlay_Scrolling .....	479
GetTextOverlay_ScrollingSpeed .....	480
GetTextOverlay_Shadow .....	480
GetTextOverlay_ShadowColor .....	480
GetTextOverlay_ShadowDirection .....	480
GetTextOverlay_String .....	481
GetTextOverlay_TargetDisplay .....	481
GetTextOverlay_Top .....	481
GetTextOverlay_Transparent .....	482
GetTranslatedCoordinates .....	482
GetTVChannelInfo .....	483
GetVideoCodec .....	483
GetVideoCompressionSettings .....	484
GetVideoControlMode .....	484
GetVideoHeightFromIndex .....	485
GetVideoSizeFromIndex .....	485
GetVideoWidthFromIndex .....	486
GetVMR9ImageAdjustmentBounds .....	486
GetVUMeterSetting .....	487
GraphState .....	488
IsAudioDeviceASoundCard .....	488
IsAudioDeviceConnected .....	488
IsAudioRendererConnected .....	489
IsCameraControlSettingAvailable .....	489
IsDialogAvailable .....	490
IsDVDevice .....	490
IsPlaylistActive .....	491

IsURLResponding .....	491
IsURLVideoStreamAvailable .....	491
IsVideoControlModeAvailable .....	492
IsVideoDeviceConnected .....	492
IsVideoQualitySettingAvailable .....	493
IsVideoSignalDetected .....	493
IsVMR9ImageAdjustmentAvailable .....	494
LoadCompressorSettingsFromDataString .....	495
LoadCompressorSettingsFromTextFile .....	495
MixAudioSamples .....	496
Mixer_Activation .....	496
Mixer_AddToMixer .....	497
Mixer_RemoveFromMixer .....	499
Mixer_SetOverlayRoundedCorner .....	499
Mixer_SetupPIPFromSource .....	499
Monitor_Primary_Index .....	501
MonitorBounds .....	501
MonitorsCount .....	502
MotionDetector_CellColorIntensity .....	503
MotionDetector_CellMotionRatio .....	503
MotionDetector_EnumGridDialogControls .....	504
MotionDetector_Get2DTextGrid .....	505
MotionDetector_Get2DTextMotion .....	505
MotionDetector_GetCellLocation .....	506
MotionDetector_GetCellSensitivity .....	507
MotionDetector_GetCellSize .....	507
MotionDetector_GlobalColorIntensity .....	508
MotionDetector_GloballyIncOrDecSensitivity .....	508
MotionDetector_Reset .....	509
MotionDetector_ResetGlobalSensitivity .....	509
MotionDetector_SetCellSensitivity .....	510
MotionDetector_SetGridSize .....	511
MotionDetector_ShowGridDialog .....	511
MotionDetector_TriggerNow .....	512
MotionDetector_UseThisReferenceSample .....	512
MP4NeedsReindexing .....	513
MPEGProgramSetting .....	514
MultipurposeEncoder_QuickConfigure_UDPStreaming_H264 .....	514
MultipurposeEncoder_ReindexClip .....	515
NotifyPlayerTrackbarAction .....	515
ONVIF_GetStr .....	516
ONVIF_SetStr .....	516
ONVIFCancelDiscovery .....	517
ONVIFDeviceInfo .....	517
ONVIFDiscoverCameras_IPRange .....	518
ONVIFDiscoverCameras_Multicast .....	518
ONVIFEnumCamerasDiscovered .....	519
ONVIFPTZGetLimits .....	519
ONVIFPTZGetPosition .....	520
ONVIFPTZPreset .....	520

ONVIFPTZSendAuxiliaryCommand .....	521
ONVIFPTZSetPosition .....	521
ONVIFPTZStartMove .....	522
ONVIFPTZStopMove .....	522
ONVIFSnapShot .....	523
OpenDVD .....	523
OpenPlayer .....	524
OpenPlayerAtFramePositions .....	525
OpenPlayerAtTimePositions .....	526
OpenURLAsyncStatus .....	527
PausePlayer .....	527
PausePreview .....	528
PauseRecording .....	529
PlayerFrameStep .....	529
Playlist .....	530
PointGreyConfig .....	530
PutMiscDeviceControl .....	531
RecordingKBytesWrittenToDisk .....	532
RecordToNewFileNow .....	532
RefreshDevicesAndCompressorsLists .....	533
RefreshPlayerOverlays .....	534
ResetVideoDeviceSettings .....	534
ResumePreview .....	535
ResumeRecording .....	535
RetrieveInitialXYAfterRotation .....	536
RewindPlayer .....	537
RunPlayer .....	538
RunPlayerBackwards .....	538
SaveCompressorSettingsToDataString .....	539
SaveCompressorSettingsToTextFile .....	540
ScreenRecordingUsingCoordinates .....	540
SendCameraCommand .....	541
SendDVCommand .....	542
SendImageToVideoFromBitmaps .....	542
SendImageToVideoFromBitmaps2 .....	543
SendIPCameraCommand .....	544
SetAudioRendererAdditional .....	544
SetAuthentication .....	545
SetAVIMuxConfig .....	545
SetCameraControl .....	546
SetCameraExposure .....	547
SetDecryptionKey .....	547
SetDisplayActive .....	547
SetDisplayAlphaBlendEnabled .....	548
SetDisplayAlphaBlendValue .....	548
SetDisplayAspectRatio .....	548
SetDisplayAutoSize .....	549
SetDisplayEmbedded .....	549
SetDisplayFullScreen .....	550
SetDisplayHeight .....	550

SetDisplayLeft .....	550
SetDisplayLocation .....	551
SetDisplayMonitor .....	551
SetDisplayMouseMovesWindow .....	551
SetDisplayPanScanRatio .....	552
SetDisplayParent .....	552
SetDisplayStayOnTop .....	553
SetDisplayTop .....	553
SetDisplayTransparentColorEnabled .....	553
SetDisplayTransparentColorValue .....	554
SetDisplayVideoPortEnabled .....	554
SetDisplayVisible .....	554
SetDisplayWidth .....	555
SetEncryptionKey .....	555
SetFFmpegAudioFilter .....	555
SetFFmpegFilter .....	556
SetFrameCaptureBounds .....	556
SetFWCam1394 .....	557
SetHeaderAttribute .....	558
SetImageOverlay_AlphaBlend .....	560
SetImageOverlay_AlphaBlendValue .....	560
SetImageOverlay_Attributes .....	561
SetImageOverlay_Attributes2 .....	563
SetImageOverlay_ChromaKey .....	563
SetImageOverlay_ChromaKeyLeewayPercent .....	564
SetImageOverlay_ChromaKeyRGBColor .....	565
SetImageOverlay_Enabled .....	566
SetImageOverlay_Height .....	566
SetImageOverlay_LeftLocation .....	567
SetImageOverlay_RotationAngle .....	568
SetImageOverlay_StretchToVideoSize .....	569
SetImageOverlay_TargetDisplay .....	570
SetImageOverlay_TopLocation .....	571
SetImageOverlay_Transparent .....	572
SetImageOverlay_TransparentColorValue .....	573
SetImageOverlay_UseTransparentColor .....	573
SetImageOverlay_Width .....	574
SetImageOverlayFromBMPFile .....	575
SetImageOverlayFromBMPFile2 .....	576
SetImageOverlayFromHBitmap .....	576
SetImageOverlayFromHBitmap2 .....	577
SetImageOverlayFromImageFile .....	578
SetImageOverlayFromImageFile2 .....	579
SetImageOverlayFromJPEGFile .....	580
SetImageOverlayFromJPEGFile2 .....	581
SetImageOverlayFromTBitmap .....	581
SetImageOverlayFromTBitmap2 .....	582
SetImageOverlayFromTImage .....	583
SetImageOverlayFromTImage2 .....	584
SetIPCameraSetting .....	584

SetLocation .....	585
SetLogoFromBMPFile .....	586
SetLogoFromHBitmap .....	586
SetLogoFromJPEGFile .....	587
SetLogoFromTBitmap .....	587
SetLogoFromTImage .....	588
SetMultiplexerFilterByName .....	588
SetParentWindow .....	589
SetTextOverlay_Align .....	589
SetTextOverlay_AlphaBlend .....	590
SetTextOverlay_AlphaBlendValue .....	590
SetTextOverlay_BkColor .....	591
SetTextOverlay_CustomVar .....	592
SetTextOverlay_Enabled .....	593
SetTextOverlay_Font .....	594
SetTextOverlay_FontColor .....	595
SetTextOverlay_GradientColor .....	596
SetTextOverlay_GradientMode .....	597
SetTextOverlay_HighResFont .....	597
SetTextOverlay_Left .....	598
SetTextOverlay_Right .....	599
SetTextOverlay_Scrolling .....	600
SetTextOverlay_ScrollingSpeed .....	601
SetTextOverlay_Shadow .....	602
SetTextOverlay_ShadowColor .....	602
SetTextOverlay_ShadowDirection .....	603
SetTextOverlay_String .....	604
SetTextOverlay_TargetDisplay .....	605
SetTextOverlay_Top .....	606
SetTextOverlay_Transparent .....	607
SetVideoCompressionDefaults .....	607
SetVideoCompressionSettings .....	608
SetVideoControlMode .....	609
SetVideoControlMode2 .....	609
SetVideoQuality .....	610
SetVMR9ImageAdjustmentValue .....	610
SetVuMeter_Enabled .....	611
SetVUMeterSetting .....	611
SetWindowRecordingByHandle .....	612
SetWindowRecordingByName .....	612
SetWindowTransparency .....	613
ShapeOverlayList .....	614
ShowDebugWindow .....	615
ShowDialog .....	615
StartAudioRecording .....	616
StartAudioRendering .....	617
StartPreview .....	618
StartRecording .....	618
StartReencoding .....	619
StartSynchronized .....	619



Stop .....	620
StopPlayer .....	621
StopPreview .....	621
StopRecording .....	622
StopReencoding .....	622
StreamInterface_PushData .....	623
TextOverlay_CreateCustomFont .....	623
TextOverlay_CreateCustomFont2 .....	624
ThirdPartyFilter_AddToList .....	625
ThirdPartyFilter_ClearList .....	626
ThirdPartyFilter_Enable .....	626
ThirdPartyFilter_RemoveFromList .....	627
ThirdPartyFilter_ShowDialog .....	627
TVClearFrequencyOverrides .....	628
TVGetMinMaxChannels .....	628
TVSetChannelFrequencyOverride .....	629
TVStartAutoScan .....	629
TVStopAutoScan .....	630
UseNearestVideoSize .....	630
v360_AddYawPitchRoll .....	631
v360_GetAngle .....	631
v360_GetYawPitchRoll .....	632
v360_ResetAnglesToDefault .....	632
v360_SetAngle .....	633
v360_SetInterpolation .....	633
v360_SetProjection .....	633
v360_SetStereoFormat .....	634
v360_SetTranspose .....	634
v360_SetYawPitchRoll .....	635
VideoCompressorIndex .....	635
VideoDeviceIndex .....	636
VideoDeviceIndexFromId .....	636
VideoFormatIndex .....	637
VideoFromImages_CreateSetOfBitmaps .....	637
VideoInputIndex .....	638
VideoQualityAuto .....	638
VideoQualityDefault .....	639
VideoQualityMax .....	639
VideoQualityMin .....	640
VideoQualityStep .....	640
VideoQualityValue .....	640
VideoSizeIndex .....	641
VideoSubtypeIndex .....	641
WriteScriptCommand .....	642
Events .....	642
OnAudioBufferNegotiation .....	644
OnAudioDeviceSelected .....	644
OnAudioPeak .....	645
OnAuthenticationNeeded .....	645
OnAVIDurationUpdated .....	646

OnBacktimedFramesCountReached .....	647
OnBitmapsLoadingProgress .....	648
OnClick .....	648
OnClientConnection .....	648
OnColorKeyChange .....	649
OnCopyPreallocDataCompleted .....	650
OnCopyPreallocDataProgress .....	650
OnCopyPreallocDataStarted .....	651
OnCreatePreallocFileCompleted .....	651
OnCreatePreallocFileProgress .....	652
OnCreatePreallocFileStarted .....	652
OnDbClick .....	652
OnDeviceArrivalOrRemoval .....	653
OnDeviceLost .....	653
OnDeviceReconnected .....	654
OnDeviceReconnecting .....	654
OnDirectNetworkStreamingHostUrl .....	655
OnDiskFull .....	655
OnDragDropFiles .....	656
OnDVCommandCompleted .....	656
OnDVDDiscontinuity .....	657
OnEnumerateWindows .....	657
OnFilterSelected .....	658
OnFrameBitmap .....	658
OnFrameCaptureCompleted .....	660
OnFrameOverlayUsingDC .....	661
OnFrameOverlayUsingDIB .....	662
OnFrameOverlayUsingVIDEOHDR .....	663
OnFrameProgress .....	664
OnFrameProgress2 .....	665
OnGraphBuilt .....	665
OnInactive .....	666
OnKeyPress .....	666
OnLastCommandCompleted .....	667
OnLeavingFullScreen .....	667
OnLog .....	668
OnMotionDetected .....	668
OnMotionNotDetected .....	669
OnMouseDown .....	669
OnMouseDown_Video .....	671
OnMouseDown_Window .....	671
OnMouseMove .....	671
OnMouseMove_Video .....	673
OnMouseMove_Window .....	673
OnMouseUp .....	673
OnMouseUp_Video .....	675
OnMouseUp_Window .....	675
OnMouseWheel .....	675
OnNoVideoDevices .....	676
OnNTPTimestamp .....	677

OnONVIFDiscoveryCompletedNotification .....	677
OnOpenURLAsyncStatusChanged .....	678
OnPlayerBufferingData .....	678
OnPlayerEndOfPlaylist .....	679
OnPlayerEndOfStream .....	679
OnPlayerOpened .....	680
OnPlayerStateChanged .....	680
OnPlayerUpdateTrackbarPosition .....	681
OnPreviewStarted .....	681
OnRawAudioSample .....	682
OnRawVideoSample .....	682
OnRecordingCompleted .....	683
OnRecordingPaused .....	683
OnRecordingReadyToStart .....	684
OnRecordingStarted .....	685
OnReencodingCompleted .....	686
OnReencodingProgress .....	686
OnReencodingStarted .....	687
OnReinitializing .....	687
OnResizeVideo .....	687
OnTextOverlayScrollingCompleted .....	688
OnThirdPartyFilterConnected .....	688
OnThreadSync .....	689
OnTVChannelScanCompleted .....	689
OnTVChannelScanStarted .....	690
OnTVChannelSelected .....	690
OnVideoCompressionSettings .....	691
OnVideoDeviceSelected .....	692
OnVideoFromBitmapsNextFrameNeeded .....	692
Types .....	693
TAero .....	693
TApplicationPriority .....	693
TASFDeinterlaceMode .....	694
TASFProfileVersion .....	694
TAspectRatio .....	694
TAudioSource .....	695
TAuthenticationType .....	695
TAutoFileName .....	696
TAVIInfoType .....	697
TAVIMuxConfig .....	697
TCardinalDirection .....	698
TDialog .....	699
TDiscoveryCallbackStatus .....	699
TDVCommand .....	700
TDVDInfoType .....	701
TDVSize .....	701
TDVVideoFormat .....	702
TDVVideoStandard .....	702
TEncoder_int .....	703
TEncoder_str .....	704

TEncryptionMethod .....	704
TEventNotification .....	705
TFileSort .....	705
TFormatType .....	705
TFrameBitmapInfo .....	706
TFrameCaptureDest .....	706
TFrameInfo .....	707
TFrameInfoId .....	708
TFrameInfoStringId .....	709
TGPUEncoder .....	709
TGraphState .....	709
THeaderAttribute .....	710
THwAccel .....	710
TIPCameraSetting .....	710
TLogoLayout .....	711
TLogType .....	711
TMiscDeviceControl .....	716
TMouseButton .....	716
TMPEGProgramSetting .....	716
TMpegStreamType .....	717
TMultiplexedRole .....	717
TNDIFormatType .....	718
TNetworkStreaming .....	718
TNetworkStreamingType .....	719
TNotificationMethod .....	719
TOnAudioBufferNegotiation .....	720
TOnAudioPeak .....	720
TOnAuthenticationNeeded .....	720
TOnAVIDurationUpdated .....	721
TOnClientConnection .....	721
TOnColorKeyChange .....	722
TOnCreatePreallocatedFileCompleted .....	722
TOnDeviceArrivalOrRemoval .....	722
TOnDirectNetworkStreamingHostUrl .....	723
TOnDragDropFiles .....	724
TOnDVCommandCompleted .....	724
TOnDVDDiscontinuity .....	724
TOnEnumerateWindows .....	725
TOnFileNotification .....	725
TOnFilterSelected .....	725
TOnFrameCaptureCompleted .....	726
TOnFrameOverlayUsingDC .....	726
TOnFrameOverlayUsingDIB .....	727
TOnFrameProgress .....	729
TOnLog .....	729
TOnMotionDetected .....	730
TOnMotionNotDetected .....	730
TOnMouseWheel .....	731
TOnONVIFDiscoveryCompletedNotification .....	731
TOnOpenURLAsyncStatusChanged .....	732

TOnPlayerBufferingData .....	732
TOnPlayerStateChanged .....	733
TOnProgress .....	733
TOnProgressCommented .....	734
TOnRawSample .....	734
TOnRecordingCompleted .....	735
TOnRecordingReadyToStart .....	736
TOnResizeVideo .....	736
TOnSourceFileToDestFileCompleted .....	737
TOnSourceFileToDestFileStarted .....	737
TOnTextOverlayScrollingCompleted .....	738
TOnThirdPartyFilterConnected .....	738
TOnThreadSync .....	739
TOnTVChannelScanStarted .....	739
TOnTVChannelSelected .....	739
TOnVideoCompressionSettings .....	740
TOnVideoFromBitmapsNextFrameNeeded .....	741
TOnVideoKeyPress .....	741
TOnVideoMouseMove .....	742
TOnVideoMouseUpDown .....	742
TONVIFDeviceInfo .....	743
TOpenURLAsyncStatus .....	744
TPlayerState .....	744
TPlaylist .....	745
TPointGreyConfig .....	745
TRawSampleCaptureLocation .....	746
TRecordingMethod .....	746
TRecordingTimer .....	747
TRegistryRoot .....	747
TRGBSelector .....	748
TStreamType .....	748
TSynchronizationRole .....	748
TSyncPreview .....	748
TTextOverlayGradientMode .....	749
TThirdPartyFilterList .....	750
TThreadSyncPoint .....	751
TTrackbarAction .....	751
TTriState .....	751
TTunerMode .....	752
TTVChannelInfo .....	752
Tv360_Angle .....	753
Tv360_InOut .....	753
Tv360_Interpolation .....	754
TV360_MouseAction .....	754
Tv360_Projection .....	754
Tv360_StereoFormat .....	756
TVideoAlignment .....	756
TVideoDeinterlacing .....	756
TVideoRenderer .....	757
TVideoRendererPriority .....	757

TVideoRotation .....	758
TVideoSource .....	758
TVMR9ImageAdjustment .....	759
TVUMeter .....	759
TVUMeterSetting .....	760
TWebcamStillCaptureButton .....	761

## General

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

---

## Online Help

### Online Help

[Prev](#)
[Next](#)

#### Datastead TVideoGrabber SDK 16.2 help Description

**The latest version of the TVideoGrabber SDK 16.2 documentation is available here:**

<https://www.datastead.com/video-sdk/>

The FAQ, including code snippets, is available here:

<https://www.datastead.com/faq/>

For a question feel free to contact us at [contact@datastead.com](mailto:contact@datastead.com) or from our contact page:

<https://www.datastead.com/contact/>

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

---

## Features

### Features

[Prev](#)
[Next](#)

#### TVideoGrabber Video SDK v16.2

for C#, VB.NET, VC++, Delphi and C++Builder (all versions), and ActiveX compatible tools (PowerBuilder, LabView,...)

©2025 Datastead Software

Home page:

<https://www.datastead.com/>

#### Description

#### CAPTURE AND PREVIEW OF VIDEO SOURCES

The TVideoGrabber SDK supports most of the video sources and capture devices:

- ONVIF cameras with ONVIF discovery PTZ support (\*)
  - RTSP cameras and servers
  - streaming URLs (\*)
  - webcams,
  - USB video capture devices,
  - PCIe video capture cards,
  - Blackmagic Decklink cards,
  - USB composite capture devices (e.g. Easycap),
  - GigE cameras (e.g. PointGrey)
- (\*) requires our optional [Datastead RTSP/RTMP/HTTP/ONVIF Source Filter](#)

#### VIDEO RECORDING

- audio/video recording to various formats

through the [Datastead Encoder](#) after installing our optional [Datastead Multipurpose Encoder filter](#) (does not require invoking the FFmpeg command line, although this is possible if preferred)

Most of the recording formats include MP4, MOV, FLV, AVI, Ogg/Theora, WebM, etc...

- **audio/video recording through third-party DirectShow filters**

- AVI recording,
- WMV/ASF recording,

- MP4/FLV recording (requires a third-party H264 or AAC DirectShow encoder)
- Timer-based recording (delayed start, timered stop, or create new file periodically)
- Compression of the audio and video streams, "on-the-fly" or after recording,
- Pause/resume during recording, with possibility to create a new clip for each pause/resume,
- Preview while recording.

## **NETWORK STREAMING**

- video/audio streaming through our optional [Datastead Multipurpose Encoder filter](#),
- MMS streaming,
- NDI streaming through our optional [Datastead NDI filters](#)

## **MEDIA PLAYER**

- play most of the audio/video clips after installing the [LAV decoder filters](#).
- play static images: BMP, PNG, JPEG, GIF, ...
- trackbar control,
- playlist support,
- playback at higher or lower speed, forward or backward,
- fast seeking, forward or backward,
- synchronization of several TVideoGrabber player components,
- play streaming sources,
- opens static JPEG, BMP, PNG, GIF image files, allowing image processings and overlays

## **FRAME CAPTURE**

- on the fly during preview, recording or playback, to memory bitmap, or to BMP, JPEG, PNG, TIFF files

## **IP CAMERAS / IP VIDEO SERVERS**

- supports RTSP / ONVIF IP cameras and other URL sources: HTTP, RTMP, UDP/MPEG-TS, RTSPS, HTTPS (requires our [Datastead RTSP/RTMP/HTTP/ONVIF Source Filter](#) )
- support for ONVIF PTZ control

## **MOTION DETECTION**

- the video frame area can be divided in cells of motion detection,

## **MULTIPLE VIDEO WINDOWS**

- assigned to different monitors (e.g. a small video window on a monitor and a full screen video on the 2nd and 3rd monitors)

## **VIDEO PROCESSING AND OVERLAYS**

- multiple text and image overlays, over the video frame or directly over the video window(s),
- rotation, resizing, alpha blending, chroma keying,
- video transparency,
- chroma key,
- video rotation 90° 180° 270°,
- top-down and left-right,
- half size and full size deinterlacing,
- brightness, contrast, saturation, hue,
- inverted colors,
- greyscale
- video processing using FFmpeg filters with [SetFFmpegFilter](#) (requires the Datastead RTSP/RTMP/HTTP/ONVIF Source filter)

## **SCREEN RECORDING**

- with or without cursor
- full screen recording (or one monitor when using several monitors),
- recording of the extended desktop across several monitors,
- recording of a given window identified by its name or handle

## **CROPPING AND ZOOMING**

- recording of only a cropped area of the video source
- zooming within areas the video source

## **VIDEO CLIP BUILT FROM BITMAPS OR IMAGE FILES (BMP, JPEG, GIF, PNG, etc...)**

- the final frame rate can be adjusted at the end of the recording

## **REENCODING OF VIDEO AND AUDIO CLIPS**

- by using the installed audio and/or video compressors,
- by applying graphics and/or text overlays,
- by extracting sequences from a start and stop time.

## **AUDIO VU-METERS**

- "needle analog" style or "digital bargraph" style

## **MULTIPLEXED INPUTS SUPPORT**



- supports video cards with several inputs,
- accept 4 switched inputs and 16 switched inputs in 2x2 or 4x4 mosaic video or in master/slave modes,

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## New features

### New features

[Prev](#)

[Next](#)

New features

#### Description

#### NEW FEATURES

The version 16.2 introduces:

- . the [MP4NeedsReindexing](#) function
- . the [SetFFmpegAudioFilter](#) function
- . the [Facebook GoLive ReturnStreamURL](#) function
- . the [OnOpenURLAsyncStatusChanged](#) event
- . the [AES encryption/decryption](#) with [SetEncryptionKey](#) and [SetDecryptionKey](#)
- . the possibility to play a .MP4 or .MKV video currently being recorded by enabling [VideoPlayableWhileRecording](#)

Note that some features above require the Datastead RTSP/RTMP/HTTP/ONVIF Source and/or the Datastead Multipurpose Encoder to be installed.

The changelog including new features and bug fixes can be found [here](#).

Created with the Standard Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

## Development platforms - System requirements

### Development platforms - System requirements

[Prev](#)

[Next](#)

Development platforms - System requirements

#### Description

#### DEVELOPMENT TOOLS SUPPORTED

- Delphi and C++Builder
- Visual Studio .NET
- C++, QT
- ActiveX

#### SYSTEM REQUIREMENTS

Windows 11, Windows 10, Windows 8.1, Windows 8, Windows 7, Windows XP

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

## Support - Contact

### Support - Contact

[Prev](#)

[Next](#)

Product support.

#### Description

## Support

online documentation: <https://www.datastead.net/vidgrabhelp/Datastead%20TVideoGrabber%20SDK.html>  
 FAQ: <https://www.datastead.com/faq/>

## Contact emails

Support questions: [support@datastead.com](mailto:support@datastead.com)  
 Sales questions: [sales@datastead.com](mailto:sales@datastead.com)  
 Other questions: [contact@datastead.com](mailto:contact@datastead.com)

## Contact form

<https://www.datastead.com/contact/>

---

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer](#)

---

## Licensing

### Licensing

[Prev](#)

[Next](#)

Licensing

#### Description

Our license is a commercial, per-developer(\*), royalty-free license.  
 The application developed can be distributed on as many PCs, as needed without having to pay additional end user runtime fees.

(\*)

- if you are developing as contractor for another company, a license must be purchased in the name of this company.
- if several developers are working concurrently on this SDK, the corresponding number of licenses must be purchased.

The licenses can be purchased from our online store:

<http://www.datastead.com/purchase.html>

[License Agreement.](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

---

## User guide

---

Created with the Standard Edition of HelpNDoc: [Easily create CHM Help documents](#)

---

## Datastead Encoder

### Datastead Encoder

[Prev](#)

[Next](#)

Datastead Encoder

#### Description

The Datastead Encoder is a new set of codecs allowing to:

- compress video in various formats including MPEG4, MJPEG, H264, HEVC, eventually by using the GPU encoder if available (NVidia, AMD and Intel Quicksync supported),
- compress audio in various formats including AAC, Mpeg2
- record to file in various container formats (.avi, .mp4, .mkv, etc...),
- stream in various formats like UDP, RTMP, RTP, etc...

The Datastead Encoder requires the [Datastead Multipurpose Encoder](#) version 2 or newer to be installed, **however it does not launch FFmpegLGPL.exe**, nor requires to run any external process. One or more Datastead Encoder instances can be associated with a given TVideoGrabber component (e.g. an encoder instance performing the recording and another instance performing the streaming of the same TVideoGrabber instance)

A Datastead Encoder instance can record what is being displayed/rendered to a file, or send a live stream to the network.

When adding an Encoder instance, it returns an unique ID that is then used to configure this instance.

## INSTALLATION

To install it:

### - **option 1:**

run the .exe installer located in the "Install" folder of the Multipurpose Encoder package. Then TVideoGrabber will locate the binaries automatically.

### - **option 2:**

copy the x86 and x64 folders containing the encoder binaries:

- either directly under the folder where is located the application's ".exe", e.g.:

*Example of layout:*

c:/appfolder/application.exe

c:/appfolder/x86

c:/appfolder/x64

- either to another folder, that you specify with VideoGrabber.[ExtraDLLPath](#)

*Example:*

VideoGrabber.ExtraDLLPath = "c:/AnotherFolder"

*and the corresponding layout:*

c:/appfolder/application.exe

c:/AnotherFolder/x86

c:/AnotherFolder/x64

## ACTIVATION AND CONFIGURATION

**Note: the instance and parameters must be configured before starting the video.**

### Creating and configuring an instance

- to create an instance that will write to a file, invoke [Encoders.CreateInstanceForRecording](#)
- to create an instance that will stream to the network, invoke [Encoders.CreateInstanceForStreaming](#)

Both functions return an unique ID that is attributed when creating the instance.

To configure then encoder, this ID must be passed to [Encoder.SetStr](#) and [Encoder.SetInt](#).

### While the video is running:

- the instance can be paused with [Encoder.Pause](#) and resumed with [Encoder.Resume](#)
- if recording:
  - the current file being written can be closed and a new file opened by invoking [Encoder.NewOutputFile](#)
  - the current file being written can be closed by invoking [Encoder.CloseOutputFile](#)
  - the total amount of bytes written can be read with [Encoder.GetInt](#) (Enc\_Bytes\_Written\_kb\_readonly, ...)

### Once the video is stopped (VideoGrabber.Stop()):

- an encoder instance can be eventually removed by invoking [Encoders.RemoveInstance](#), otherwise it will be active again when restarting the video.
- to remove all the encoders instances associated with a TVideoGrabber component invoke

[Encoders.RemoveAllInstances](#)

### Restarting the video after stopped

If an instance is not removed it will be active again when restarting the video.

So if it is configured for recording, this will overwrite the current recording file, unless a new file is specified with [Encoder.NewOutputFile](#) before invoking StartPreview()

## RECORDING

The recording is described in [Recording through the Datastead Encoder](#)

## STREAMING

The streaming is described in [Streaming through the Datastead Encoder](#)

## CONFIGURATION OF THE ENCODER

The configuration is achieved through the [Encoder.SetInt](#) and [Encoder.SetStr](#) functions.

Example of various values that can be passed:

### - to select the codec:

Encoder.SetStr (ENCODER\_RECORDING\_ID, Enc\_Video\_Codec, "mjpeg");

Encoder.SetStr (ENCODER\_RECORDING\_ID, Enc\_Audio\_Codec, "aac");

Encoder.SetStr (ENCODER\_RECORDING\_ID, Enc\_Video\_Codec, "mpeg4");

Encoder.SetStr (ENCODER\_RECORDING\_ID, Enc\_Video\_Codec, "mpeg2");

```
Encoder_SetStr (ENCODER_RECORDING_ID, Enc_Video_Codec, "jpeg");
Encoder_SetStr (ENCODER_RECORDING_ID, Enc_Video_Codec, "h264");
Encoder_SetStr (ENCODER_RECORDING_ID, Enc_Video_Codec, "hevc");
- to configure the GPU encoder (usually for the "h264", "hevc" or "mjpeg" codecs:
Encoder_SetInt (ENCODER_RECORDING_ID, , (int) Enc_GPU_Intel_QSV);
(possible values include Enc_GPU_None, Enc_GPU_Auto, Enc_GPU_Intel_QSV,
Enc_GPU_NVidia_NVENC, Enc_GPU_AMD_AMF)
- to set the bit rate:
Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_Bitrate_kb, 2000);
- to configure the bitrate control:
Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_rc_BufferSize_kb, 1000);
Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_MinBitrate_kb, 2000);
Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_MaxBitrate_kb, 2000);
```

## See Also

[Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveAllInstances](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## Pre-requisties

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

### Pre - requisities

## Pre - requisities

[Prev](#)

[Next](#)

### Pre-requisties

#### Description

#### IP CAMERAS, URL SOURCES

To be able to decode IP cameras and URL sources, and record them without decompression/recompression(\*), the [Datastead RTSP/RTMP/HTTP/ONVIF Source filter](#) is required.

(\*) when *VideoGrabber.RecordingInNativeFormat* is enabled

#### RECORDING

Webcams, URL sources, screen, PCIe HD capture cards, video mixer:

To be able to record by using the Datastead codecs (h264, hevc, etc...), including the NVidia/AMD/Intel GPU encoders, the [Datastead Multipurpose Encoder](#) is required (starting from the v2.01, it does not run external processes)

See the [Datastead Encoder](#) chapter for more information

#### PLAYER

For the decoding of video files we recommend to install the [LAV Filters](#) that can be downloaded here: <https://github.com/Nevcairiel/LAVFilters/releases/>

If not using the LAV Filters, note that the Microsoft H264 decoder installed on Windows may produce glitches/pixellation when running from the debugger. However there is no problem out of the debugger.

#### INSTALLATION

All the filters above can be installed:

**A) either by running the .exe the installer**

**B) either by copying the x86 and x64 folders directly under the folder where is located the application's .exe:**

Example of file layout:

c:/appfolder/application.exe

c:/appfolder/x86

c:/appfolder/x64

where x86 and x64 are the folders containing the respective LAV filters binaries.

In this case TVideoGrabber will find them automatically.

**C) either by copying the x86 and x64 folders to another folder, that is specified in the VideoGrabber.ExtraDLLPath property:**

Example:  
 VideoGrabber.ExtraDLLPath = c:/**codec**folder  
 File layout:  
 c:/appfolder/application.exe  
 c:/**codec**folder/x86  
 c:/**codec**folder

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## Platform-dependent properties

Created with the Standard Edition of HelpNDoc: [Make Help Documentation a Breeze with a Help Authoring Tool](#)

### Platform dependent properties

#### Platform dependent properties

[Prev](#)

[Next](#)

List of platform-dependent properties.

#### Description

The functions below return platform-dependent information.

[VideoDevices](#) : list of the video capture devices available

[VideoDevicesCount](#) : count of the current video capture devices available

[AudioDevices](#) : list of the current audio capture devices available

[AudioDevicesCount](#) : count of the current audio capture devices available

[VideoCompressors](#) : list of the current video encoders available

[VideoCompressorsCount](#) : count of the current video encoders available

[AudioCompressors](#) : list of the current audio encoders available

[AudioCompressorsCount](#) : count of the current audio encoders available

**TVideoGrabber exposes the following propeties, that are the indexes (in the lists above) of the current devices or compressors used by the TVideoGrabber component :**

[VideoDevice](#) : selects the video capture device to be used during preview or recording

[AudioDevice](#) : selects the audio capture device to be used during preview or recording

[VideoCompressor](#) : selects the video compressor to be used during recording

[AudioCompressor](#) : selects the audio compressor to be used during recording

[VideoDeviceName](#) : returns the name of the current video capture device

[AudioDeviceName](#) : returns the name of the current audio capture device

[VideoCompressorName](#) : returns the name of the current video compressor

[AudioCompressorName](#) : returns the name of the current audio compressor

Created with the Standard Edition of HelpNDoc: [Quickly and Easily Convert Your Word Document to an ePub or Kindle eBook](#)

## Device-dependent properties

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

### Device dependent properties

#### Device dependent properties

[Prev](#)

[Next](#)

Explanations about device-dependent properties.

### Description

Most of the video capture device properties are device-dependent.

This means that their content changes when the current video capture device changes.

For this reason these properties are not published, because setting a value from the Object Inspector does allow to specify the concerned video device (selected by the [VideoDevice](#) property).

TVideoGrabber saves and retrieves these values for each capture device in the registry in HKEY\_CURRENT\_USER, under the key. if the "Datastead" key is destroyed it will be recreated automatically and the device settings will be reset to their default values.

The content of these properties is refreshed when the [OnVideoDeviceSelected](#) event occurs after selecting the video capture device by assigning a new value to [VideoDevice](#) .

The principle is the same for the audio device ([OnAudioDeviceSelected](#) and [AudioDevice](#) ).

E.g., we have 2 video capture devices on our platform.

- the 1st video device has 3 inputs: "Tuner", "Composite" and "SVideo",
- the 2nd device has 2 composite inputs: "Composite 0" and "Composite 1".

a) on the first video capture device (VideoDevice = 0), the VideoInputs string list returns the following values: "Tuner", "Composite", "SVideo"

The index value VideoInput = 1 means that the Composite video input has been selected on this video capture device.

b) on the 2nd video capture device (VideoDevice = 1), the VideoInputs string list returns the following values: "Composite 0", "Composite 1".

Now the index value VideoInput = 1 means that the "Composite 1" input has been selected on this video capture device.

Moreover, a VideoInput = 2 value that means "SVideo" on the 1st device has no meaning and is out of range on the 2nd device.

Thus, the edit fields, listboxes and indexes of any application using TVideoGrabber MUST be refreshed when the [OnVideoDeviceSelected](#) or [OnAudioDeviceSelected](#) event occurs, otherwise their content will no longer reflect the current values of the TVideoGrabber properties.

We find mainly 2 types of properties:

- single values
- list / index based values, that can be easily identified as they are composed of a string list, an index in the list, and a list count. e.g.: VideoInput, VideoInputs and VideoInputsCount.

The main device-dependent properties related to the video capture device are:

[AnalogVideoStandard](#)  
[AnalogVideoStandards](#)  
[AnalogVideoStandardsCount](#)  
[CameraControlAuto](#)  
[FrameRate](#)  
[IsAnalogVideoDecoderAvailable](#)  
[IsCameraControlAvailable](#)  
[IsDigitalVideoIn](#)  
[IsHorizontalSyncLocked](#)  
[IsTimeCodeReaderAvailable](#)  
[IsTVAudioAvailable](#)  
[IsTVTunerAvailable](#)  
[IsDVCommandAvailable](#)  
[IsVideoControlAvailable](#)  
[IsVideoCrossbarAvailable](#)  
[IsVideoInterlaced](#)  
[IsVideoPortAvailable](#)

[IsVideoQualityAvailable](#)  
[IsWDMVideoDriver](#)  
[VCRHorizontalLocking](#)  
[VideoInput](#)  
[VideoInputs](#)  
[VideoInputsCount](#)  
[VideoQualityAuto](#)  
[VideoSize](#)  
[VideoSizes](#)  
[VideoSizesCount](#)  
[VideoSubtype](#)  
[VideoSubtypes](#)  
[VideoSubtypesCount](#)

The main device-dependent properties related to the audio capture device are:

[AudioInputBalance](#)  
[AudioInputLevel](#)  
[AudioInput](#)  
[AudioInputs](#)  
[AudioInputsCount](#)  
[IsAudioCrossbarAvailable](#)

---

Created with the Standard Edition of HelpNDoc: [Simplify Your Help Documentation Process with a Help Authoring Tool](#)

---

## Video sources

---

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)

---

### Video sources supported for preview and recording

#### Video sources supported for preview and recording

[Prev](#)

[Next](#)

Video sources supported for preview and recording

##### Description

##### Video sources supported for preview and recording

The [VideoSource](#) property selects the video source to use:

##### vs\_VideoCaptureDevice:

selects the video capture devices available on the current platform (by default), then the video capture device is selected by the [VideoDevice](#) property,

##### vs\_VideoFileOrURL:

selects a file, URL or a static playlist specified by the [VideoSource FileOrURL](#) property, for preview or recording (for playback with trackbar handling use [OpenPlayer](#) and the [Player](#) functions).

##### vs\_ScreenRecording:

selects the screen as video source (to perform screen recording),

##### vs\_Mixer:

In this mode the component mixes several other TVideoGrabber components. See [How to mix several video sources into one a single one](#)

##### vs\_JPEGsOrBitmaps:

the video source are bitmap handles, or BMP files or JPEG files of the same format passed to the [OnVideoFromBitmaps NextFrameNeeded](#) event.

See [Video clips built on the fly by passing bitmap handles, BMP or JPEG files.](#)



**vs\_VideoFromImages:**

the video source is built from a set of bitmaps (BMP or JPEG files). See [Video clip from bitmaps: Overview](#).

**See Also**

[VideoSource](#) [VideoSources](#) [VideoSourcesCount](#) [TVideoSource](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

## USB and PCI video capture devices

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

### Video capture devices

#### Video capture devices

[Prev](#)

[Next](#)

Video capture devices overview.

#### Description

#### Selecting a video capture device

First of all set [VideoSource](#) = **vs\_VideoCaptureDevice** to select the video capture devices as video source.

Then select the current video capture device by assigning the [VideoDevice](#) property, which is an index in the [VideoDevices](#) property, that contains [VideoDevicesCount](#) items.

It is possible to select the video capture device programmatically by its name (as it appears in the [VideoDevices](#) list), e.g.:

```
VideoGrabber.VideoDevice = VideoGrabber.VideoDeviceIndex ("Microsoft DV Camera and VCR -
```

The name of the current video capture device is reported by [VideoDeviceName](#) .

When a video capture device is selected, its [device-dependent properties](#) are reloaded from the registry and the [OnVideoDeviceSelected](#) event occurs.

*Note: these registry settings can be reset to their default values by invoking [ResetVideoDeviceSettings](#) while the component is inactive. This is rarely needed, this may be useful if the current settings shows a black video.*

#### Identifying the video capture device by its hardware identifier (instead of its manufacturer's name)

When using several video capture devices of the same brand and model that can't be distinguished in the VideoDevices list, use [VideoDeviceIndexFromId](#) that returns the index of the device in the [VideoDevicesId](#) list.

E.g.:

```
VideoGrabber.VideoDevice = VideoGrabber.VideoDeviceIndexFromId
( "devicepnpusbvid046dpid0826mi02739103c9a0000265e8773d-8f56-11d0-a3b9-
00a0c9223196globalHD Webcam C525R" );
```

#### Connecting/disconnecting a video capture device

When a video capture device is connected or disconnected, the [OnDeviceArrivalOrRemoval](#) event occurs. This event reports the name and index of the video capture device in the [VideoDevices](#) list.

After the video capture device has been disconnected, [IsVideoDeviceConnected](#) returns true, until the device is reconnected.

If the related device is used by preview or recording, the [OnDeviceLost](#) event occurs when the device is disconnected, and the preview or recording stops.



When a video capture device is disconnected, it is not removed from the [VideoDevices](#) list, in order to prevent the complexity of having to manage the shifted indexes of the video captures whose indexes are above the index of the device currently removed (because VideoDevice selects the current video device in the VideoDevices list).

When TVideoGrabber is running, the rules are the following:

1. when a device is turned on, the device is added at the bottom of the VideoDevices list. The OnDeviceArrivalOrRemoval event occurs. Its IsDeviceArrival parameter reports "true", and its DeviceIndex parameter reports the index of the video device added in the list.
2. when a device is turned off, the device remains in the VideoDevices list. The OnDeviceArrivalOrRemoval event occurs. Its IsDeviceArrival parameter reports "false", and its DeviceIndex parameter reports the index of the video device turned off.
3. if a device turned off (case 2. above) is turned on again, the VideoDevices list does not change. The OnDeviceArrivalOrRemoval event occurs. Its IsDeviceArrival parameter reports "true", and its DeviceIndex parameter reports the index the video device had when it was turned off.

You can retrieve current the state of any video capture devices (connected or not) by testing [IsVideoDeviceConnected](#) (DeviceIndex).

E.g. IsVideoDeviceConnected (3) will return "true" if the device having the index 3 in the VideoDevices list is active and "false" if the device has been turned off.

### PID and VID identifiers of USB video capture devices

The PID and VID identifiers of the USB video capture devices are included in the [VideoDevicesId](#) list that has the same number of items than the [VideoDevices](#) list (the number of items returned by [VideoDevicesCount](#))

You can retrieve them as follows:

- assign the [VideoDevicesId](#) list to a StringList (e.g. IdList)
- read the IdList[VideoGrabber.VideoDevice] string value
- if the device is USB you will find the "pid..." and "vid..." in the string returned.

### WDM vs VFW drivers

More information [here](#).

### See Also

[WDM drivers](#) [TOnDeviceArrivalOrRemoval](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnDeviceArrivalOrRemoval](#) [OnDeviceLost](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

---

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

---

## Preview

### Preview

[Prev](#)
[Next](#)

Live preview overview.

#### Description

#### Preview of video capture devices

##### 1. select the video source

First of all, set [VideoSource](#) = vs\_ **VideoCaptureDevice** to select the video capture devices as video source

## 2. select the video capture device (optional if you have only one video capture device)

By default the 1st video capture in the [VideoDevices](#) list will be used ([VideoDevice](#) index = 0).

So you may have to select the video capture device by assigning the [VideoDevice](#) property with the index of the video capture device in the [VideoDevices](#) list.

If you need to select it by its name, [VideoDeviceIndex](#) ("...name of your video capture device...") will return the index of the name in the [VideoDevices](#) list.

E.g.

VideoGrabber.VideoDevice = VideoGrabber.VideoDeviceIndex ("...name of your video capture device...")

## 3. Start / stop / pause / resume preview

Preview can be started by using [StartPreview](#) and stopped by using [StopPreview](#) .

After starting preview, it can be paused with [PausePreview](#), and resumed with [ResumePreview](#).

## Frame progress

During preview, information about each video frame is reported by the [OnFrameProgress](#) event.

It is possible to get information about the current frame by invoking [GetFrameInfo](#) or [GetFrameInfoString](#).

## Frame capture

Frames can be captured to TBitmap, BMP files or JPEG files, one by one or automatically by using the [frame grabber](#) . See the [Frame capture](#) chapter for more information.

## Graphics and text overlays

It is possible to draw text, shapes, bitmaps or graphics over video frames. See the [Graphics and text overlays](#) chapter.

## Display options

See [Video window](#), [Dual display](#), [Transparency](#), [Third-party video renderer filter](#)

---

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

---

## Video quality

### Video quality

[Prev](#)

[Next](#)

Video quality overview.

### Description

The video quality settings allow to set brightness, hue, saturation, etc...

They are available on the [current video capture device](#) whether [IsVideoQualityAvailable](#) returns true.

The possible video quality settings are enumerated by the [TVideoQuality](#) type.

For each video quality setting, it is possible to :

- retrieve the current value with [VideoQualityValue](#)
- retrieve the "auto" mode with [VideoQualityAuto](#)
- retrieve the default value with [VideoQualityDefault](#)
- retrieve the minimum value with [VideoQualityMin](#)
- retrieve the maximum value with [VideoQualityMax](#)
- retrieve the stepping value with [VideoQualityStep](#)
- set a new value with [SetVideoQuality](#).

### Use of the [SetVideoQuality](#) parameters:

- if **SetAuto** is true, SetDefault and SetValue are ignored and the setting is switched in "auto" mode

- if **SetAuto** is false and **SetDefault** is true, SetValue is ignored and the setting is set to its default value,
- if **SetAuto** is false and **SetDefault** is false, the **SetValue** value is applied to the setting.

The default, minimum, maximum and stepping values are usually used to set trackbar ranges. See *the miscVideoQuality form of the MainDemo project for sample code*.

Note: these properties are [device-dependent](#) . Any form that uses these properties should be refreshed when the [OnVideoDeviceSelected](#) event occurs.

Look at the MainDemo project for sample code.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#)  
[VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#)  
[VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## Camera control

### Camera control

[Prev](#)
[Next](#)

Camera control overview.

#### Description

#### General camera control settings

The camera control settings allow to control pan, tilt, roll, zoom, exposure, iris, etc...  
 They are available on the [current video capture device](#) whether [IsCameraControlAvailable](#) returns true.

The possible camera controls settings are enumerated by the [TCameraControl](#) type.

For each camera control setting, it is possible to :

- retrieve the current value with [CameraControlValue](#)
- retrieve the "auto" or "manual" mode with [CameraControlAuto](#)
- retrieve the default value with [CameraControlDefault](#)
- retrieve the minimum value with [CameraControlMin](#)
- retrieve the maximum value with [CameraControlMax](#)
- retrieve the stepping value with [CameraControlStep](#)

#### Use of the [SetCameraControl](#) parameters:

- if **SetAuto** is true, SetDefault and SetValue are ignored and the setting is switched in "auto" mode
- if **SetAuto** is false and **SetDefault** is true, SetValue is ignored and the setting is set to its default value,
- if **SetAuto** is false and **SetDefault** is false, the **SetValue** value is applied to the setting.

The default, minimum, maximum and stepping values are usually used to set trackbar ranges. See *the miscVideoQuality form of the MainDemo project for sample code*.

Note: these properties are [device-dependent](#) . Any form that uses these properties should be refreshed when the [OnVideoDeviceSelected](#) event occurs.

See the MainDemo project for sample code.

#### Pan / Tilt position control

With some cameras it is possible to control the pan and tilt positioning by using [SendCameraCommand](#).

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## Detection of the video signal

### Detection of the video signal

[Prev](#)

[Next](#)

Detection of the video signal

#### Description

#### Detection of the video signal

Is it possible to detect if the video signal is present or not for each video frame, by invoking [IsVideoSignalDetected](#).

Created with the Standard Edition of HelpNDoc: [Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

## Blackmagic Decklink cards

### Blackmagic Decklink cards

[Prev](#)

[Next](#)

Using BlackMagic Decklink cards

#### Description

#### Using BlackMagic Decklink cards

Declink specific settings

To activate the video capture of the Decklink card, the video VideoSize and VideoSubtype indexes must be set to 0, and the format must be specified by the VideoFormat index, e.g.:

```
VideoGrabber.VideoSubtype = 0 (default)
VideoGrabber.VideoSize = 0 (default)
VideoGrabber.VideoFormat = VideoGrabber.VideoFormatIndex ("HDYC 1920x1080 25 fps")
VideoGrabber.VideoInput = VideoGrabber.VideoInputIndex ("HDMI")
VideoGrabber.StartPreview()
```

#### Remark:

If there is no image (black video window), it is possible that the VideoFormat selected is not suitable.

To determine the proper video format, proceed as follows:

- run the pre-compiled MainDemo.exe
- select the Declink card in the VideoDevices list
- set VideoSubtype and VideoSize to default
- select the video input in the "Video Inputs" list
- select the video format in the "Video formats" list
- start the preview

then try each one of the video formats available in the "Video Formats" list, until you see the video capture.

#### Playout to the decklink renderer

The [VideoRendererExternal](#) property lets you activate the external renderer on the Blackmagic Declink card.

- to activate it, set [VideoRendererExternal](#) = `vre_BlackMagic_Decklink`

- to deactivate it, set [VideoRendererExternal](#) = **vre\_None**

Note that the renderer requires a standard video size (e.g. 720x480 or 1920x1080).

By default TVideoGrabber resizes to 720x480.

To select the HD format, invoke [UseNearestVideoSize](#) (1920, 1080, true) before invoking OpenPlayer (to go back to the default setting invoke UseNearestVideoSize (0, 0, false))

*Note that the Decklink renderer requires also a standard frame rate (e.g. 25 fps or 29.97 fps). If the clip has a non-standard frame rate it may fail to play on the Declink output.*

## Specifying the Decklink renderer to use when more than one Decklink card is installed in the same PC

If more than one Decklink card is used in the same PC, specify the card number to use to the [VideoRendererExternalIndex](#) property (in the 0..n-1 range).

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

## IP Cameras and other network streaming sources

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

### The Datastead RTSP/RTMP/HTTP/ONVIF DirectShow source filter

#### The Datastead RTSP/RTMP/HTTP/ONVIF DirectShow source filter

[Prev](#)

[Next](#)

#### Datastead RTSP/RTMP/HTTP/ONVIF DirectShow source filter Description

##### Datastead RTSP/RTMP/HTTP/ONVIF DirectShow source filter

Capturing live streams and IP cameras require to install our optional [RTSP/RTMP/HTTP/ONVIF DirectShow source filter](#).

It is not necessary to register the filter, you can just unzip the filter binaries (includede in x86 and x64 subfolders), either under the application .exe folder, either in any folder specified as [ExtraDLLPath](#) (see also the [installation information](#))

This filter is able to decode live streams through most of the common live protocols (ONVIF, RTSP, RTMP, UDP, RTP, SDP, HTTP, MMS, etc...) and at the same time save the native frames received (H264, H265/HEVC, AAC, etc...) to .MP4, .FLV, .AVI, .MKV or .MOV containers without transcoding.

If a new file name is specified during the recording, it closes the previous file and opens the new file without loosing frames between the 2 files.

When used in TVideoGrabber the configuration of the RTSP filter is done by TVideoGrabber, however it is possible to control most of the specific filter parameters by adding and combining them at the end of the URL as follows:

VideoGrabber.[VideoSource](#) = vs\_IPCamera

VideoGrabber.[IPCameraURL](#) = "rtsp://192.168.5.1/live.sdp>**autoreconnect=0**"

VideoGrabber.[StartPreview](#)();

Other example, preview/recording of any ONVIF URL:

VideoGrabber.[VideoSource](#) = vs\_IPCamera

VideoGrabber.[IPCameraURL](#) = "onvif://192.168.5.1>**buffer=0>autoreconnect=0**"

VideoGrabber.[SetAuthentication](#) (at\_IPCamera, "user", "password");

VideoGrabber.[RecordingInNativeFormat](#) = true

VideoGrabber.[RecordingMethod](#) = rm\_MP4

VideoGrabber.[StartRecording](#)();

The possible optional parameters are listed in the "Parameter Identifiers" section of the help here:

<http://www.datastead.com/products/rtsprtmpsrc/documentation.html>

## ONVIF Cameras Discovery

### ONVIF Cameras Discovery

[Prev](#)

[Next](#)

Discovering ONVIF cameras

#### Description

#### ONVIF Cameras Network Discovery

It is possible to discover the ONVIF cameras in Multicast mode, by scanning a range of IP addresses, or both by invoking both functions consecutively.

The discovery is performed in the background.

#### Starting the discovery

- to discover the ONVIF cameras through a multicast request, invoke:

[ONVIFDiscoverCameras\\_Multicast](#) (timeout\_seconds)

- to discover the ONVIF cameras through a range of IP addresses, invoke:

[ONVIFDiscoverCameras\\_IPRange](#) (first\_IP, last\_IP, timeout\_seconds)

*Note: a timeout of 2 to 5 seconds should be sufficient.*

#### Stopping the discovery before its completion

- eventually invoke [ONVIFCancelDiscovery](#)() to stop the current background process, if needed.

#### Results of the discovery

- each camera discovered is notified by an event [OnONVIFDiscoveryCompletedNotification](#) (dcs\_CameraFound, int CameraCount)

- the end of the multicast discovery is notified by an event [OnONVIFDiscoveryCompletedNotification](#) (dcs\_MulticastCompleted, int CameraCount)

- the end of the IP range scanning is notified by an event [OnONVIFDiscoveryCompletedNotification](#) (dcs\_IPRangeCompleted, int CameraCount)

From any of these events it is possible to invoke [ONVIFEnumCamerasDiscovered](#) for each camera whose index is specified by the CameraIndex parameter.

When CameraIndex >= the number of cameras the function returns false

The sample code is included in the Delphi and C# MainDemo projects.

## ONVIF - Connecting to IP cameras through the ONVIF protocol

### ONVIF - Connecting to IP cameras through the ONVIF protocol

[Prev](#)

[Next](#)

Connecting to IP cameras through the ONVIF protocol

#### Description

#### Connecting to IP cameras through the ONVIF protocol

Connecting to an ONVIF camera through the ONVIF protocol (instead of directly the rtsp://... URL) give access:

- to the ONVIF information of the camera

- to the PTZ and other ONVIF features

#### Quick ONVIF connection

- to quick connect (without support of PTZ and other ONVIF features), use the "onvifurl://" prefix, e.g.:

VideoGrabber.[IPCameraURL](#) = "onvifurl://192.168.5.1:8080"

- to connect with PTZ support and other ONVIF features, use the "onvif://" prefix, e.g.:

VideoGrabber.[IPCameraURL](#) = "onvif://192.168.5.1:8080"

#### RTSPS support

To connect through the RTSPS protocol (for IP cameras that implement it), use the "onvifs://" or "onvifurl://" prefix.

- quick connect:

VideoGrabber.[IPCameraURL](#) = "onvifurl://192.168.5.1:8080"

- normal connect:

VideoGrabber.[IPCameraURL](#) = "onvifs://192.168.5.1:8080"

#### Sample code

VideoGrabber.[VideoSource](#) = vs\_IPCamera

VideoGrabber.[IPCameraURL](#) = "onvifurl://192.168.5.1:8080"

VideoGrabber.[SetAuthentication](#) (at\_IPCamera, "user", "password");

VideoGrabber.StartPreview();

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## ONVIF - Retrieving camera information

### ONVIF - Retrieving camera information

[Prev](#)

[Next](#)

Retrieving ONVIF information

#### Description

#### Retrieving ONVIF information

The various ONVIF information can be retrieved as strings by invoking [ONVIFDeviceInfo](#) ([TONVIFDeviceInfo](#)):!

E.g.:

VideoGrabber.VideoSource = vs\_IPCamera

VideoGrabber.IPCameraURL = "onvif://192.168.2.144"

VideoGrabber.SetAuthentication (at\_IPCamera, "user", "password");

string SerialNumber = VideoGrabber.ONVIFDeviceInfo (onv\_SerialNumber);

string PTZInfo = VideoGrabber.ONVIFDeviceInfo (onv\_PTZInfo);

string PTZPresets = VideoGrabber.ONVIFDeviceInfo (onv\_PTZPresets);

string XMLRecordings = VideoGrabber.ONVIFDeviceInfo (onv\_XMLReplay);

string FullXMLInfo = VideoGrabber.ONVIFDeviceInfo (onv\_XMLInfo);

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with a Help Authoring Tool](#)

## ONVIF - PTZ

### ONVIF - PTZ

[Prev](#)

[Next](#)

ONVIF PTZ

#### Description

#### ONVIF PTZ

Absolute, relative and continuous Pan / Tilt / Zoom are supported, as well as Presets.

Note that some camera support only partial PTZ features (e.g. the continuous move but not the absolute move, etc...)

All the values of the PTZ functions are expressed as "Double" values.

Most of the positioning functions below include a SpeedRatio parameter, the SpeedRatio value is usually in the 0 .. 1.0 range.

Before invoking any of the functions below, first [start the preview of the IP camera by using an onvif:// URL](#).

#### Min and max values

To retrieve the min and max values of each PTZ axis, invoke [ONVIFPTZGetLimits](#)

The PTZ current positions are also returned as text by invoking [ONVIFDeviceInfo](#) (onv\_PTZInfo)

#### Retrieving the current position

Invoke VideoGrabber.[ONVIFPTZGetPosition](#) to get the current pan, tilt, and zoom positions as double values

#### Continuous move

- to start a continuous invoke [ONVIFPTZStartMove](#), e.g.:

VideoGrabber.ONVIFPTZStartMove ("Pan", true, 0.5, 100)



- . the 1st parameter can be "Pan", "Tilt" or "Zoom",
- . the 2nd parameter specifies the direction or its opposite,
- . the 3rd parameter specifies the speed,
- . the 4th parameter specifies the duration of the continuous move. Note that some cameras do not implement it and go on moving until ONVIFPTZStopMove is invoked.
- to stop it, invoke [ONVIFPTZStopMove](#), e.g.:

VideoGrabber.ONVIFPTZStopMove ("Pan")

#### Absolute move

Invoke VideoGrabber.[ONVIFPTZSetPosition](#) (pan position, tilt position, zoom position, speed, **true**)

#### Relative move

Invoke VideoGrabber.[ONVIFPTZSetPosition](#) (relative pan position, relative tilt position, relative zoom position, speed, **false**)

#### Managing presets

The list of the existing presets is returned as a string by invoking [ONVIFDeviceInfo](#) (onv\_PTZPresets)

To manage a preset, invoke VideoGrabber.[ONVIFPTZPreset](#) (PresetAction, PresetName)

- PresetAction can be "CREATE", "REMOVE" or "GOTO"

- PresetName can be any name (however some camera support only their own predefined preset names)

Note: to create a preset, first position the PTZ at the desired location with the positioning functions above, then invoke ONVIFPTZPreset ("CREATE", presetname) to create it.

Then, when needed, invoke ONVIFPTZPreset ("GOTO", presetname) to move the camera to the desired location.

#### Specific manufacturer commands

To send a such command, invoke VideoGrabber.[ONVIFPTZSendAuxiliaryCommand](#) (Command)

The supported commands are described in the manufacturer's user guide of the IP camera

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

## ONVIF Replay

### ONVIF Replay

[Prev](#)

[Next](#)

#### ONVIF REPLAY

##### Description

#### ONVIF REPLAY

To use the ONVIF Replay:

1) get the list of the recordings available on the ONVIF DVR or IP camera:

```
VideoGrabber.VideoSource = vs_IPCamera
```

```
VideoGrabber.IPCameraURL = "onvif://192.168.2.144"
```

```
VideoGrabber.SetAuthentication (at_IPCamera, "user", "password");
```

```
string XMLRecordings = VideoGrabber.ONVIFDeviceInfo (onv_XMLReplay);
```

2) extract the URL from XMLRecordings, corresponding to the recording that must be replayed ("uri" field)

3) pass the URL extracted (\*) to IPCameraURL:

```
VideoGrabber.VideoSource = vs_IPCamera
```

```
VideoGrabber.IPCameraURL = "rtsp://..." // (*)
```

```
VideoGrabber.SetAuthentication (at_IPCamera, "user", "password");
```

```
VideoGrabber.StartPreview();
```

Note: if a seeking is required, add ">starttime=nnn" at the end of the replay URL (with nnn expressed in milliseconds). E.g. to start the replay at 15 seconds (15000 ms):

```
rtsp://192.168.1.22/rtsp_tunnel?rec=1&vcd=2>starttime=15000
```

Example of XML data structure returned for 2 recordings:

```
<recordings>
```

```
<recording>
```

```
<id>Cam1Rec0</id>
```

```
<source>SRC_1_0</source>
```

```
<name>_____</name>
```

```
<description>Cam1Rec0</description>
```

```
<address>LOCAL_SRC_1_0</address>
```

```
<uri>rtsp://192.168.8.139/rtsp_tunnel?
```



```
rec=1&vcd=2&enableaudio=1&audio_mode=0&aacOut=1&silence=0</uri>
</recording>
<recording>
<id>Cam1Rec1</id>
<source>SRC_1_1</source>
<name>_____</name>
<description>Cam1Rec1</description>
<address>LOCAL_SRC_1_1</address>
<uri>rtsp://192.168.8.139/rtsp_tunnel?
rec=1&inst=2&vcd=2&enableaudio=1&audio_mode=0&aacOut=1&silence=0</uri>
</recording>
</recordings>
```

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

## ONVIF snapshot

### ONVIF snapshot

[Prev](#)

[Next](#)

#### ONVIF snapshots

##### Description

#### ONVIF snapshots

To get just JPEG image from an IP camera, instead of starting the camera live stream, it is possible to request a JPEG snapshot, that is returned as a JPEG file or as a pointer to the JPEG memory buffer, or both.

Sample code to get a JPEG file:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.OpenURLAsync = true; // so the connection does not block the main thread (it's the default anyway)
VideoGrabber.IPCameraURL = "onvif://192.168.0.25:80" // if the port is not specified the default port 80 is used
VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin")
VideoGrabber.ONVIFSnapshot (false, true, "c:/folder/snapshot.jpg")
```

Sample code to get an access to the JPEG snapshot buffer in memory:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.OpenURLAsync = true; // default, so the connection does not block the main thread
VideoGrabber.IPCameraURL = "onvif://192.168.0.25:80" // if the port is not specified the default port 80 is used
VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin")
VideoGrabber.OnRawVideoSample = OnRawVideoSampleCallback
VideoGrabber.ONVIFSnapshot (true, false, "")
```

in the OnRawVideoSampleCallback event callback, the **pSampleBuffer** return a pointer to the JPEG snapshot of a **SampleBufferSize** size.

Note: it is possible to get both the memory callback and the JPEG file as follows:

```
VideoGrabber.ONVIFSnapshot (true, true, "c:/folder/snapshot.jpg")
```

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## Preview and recording of IP cameras and URL sources

### Preview and recording of IP cameras and URL sources

[Prev](#)

[Next](#)

#### Preview and recording of IP cameras and URL sources

##### Description

#### Preview and recording of IP cameras and URL sources

Note: with IP cameras supporting the ONVIF protocol no need to know RTSP URL syntax anymore, you

can open the RTSP URL by specifying the onvif:// protocol following by the IP address or domain name and port

Preview only:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.OpenURLAsync = true; // default, so the connection does not block the main thread
VideoGrabber.IPCameraURL = "onvif://192.168.0.25:80" // if the port is not specified the default port 80 is used
VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin")
VideoGrabber.AudioDeviceRendering = true // optional, if audio rendering is needed
VideoGrabber.FrameGrabber = fg_Disabled // if no overlay or frame capture is required, saves CPU
VideoGrabber.StartPreview()
```

Preview and recording:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.OpenURLAsync = true; // default, so the connection does not block the main thread
VideoGrabber.IPCameraURL = "onvif://192.168.0.25:80" // if the port is not specified the default port 80 is used
VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin")
VideoGrabber.FrameGrabber = fg_Disabled // if no overlay or frame capture is required, saves CPU
VideoGrabber.RecordingInNativeFormat = true
VideoGrabber.RecordingMethod = rm_MP4
VideoGrabber.AudioRecording = true // optional, if the URL has audio and audio recording is needed
VideoGrabber.StartRecording()
Recording only (no decoding, saves CPU):
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.OpenURLAsync = true; // default, so the connection does not block the main thread
VideoGrabber.IPCameraURL = "onvif://192.168.0.25:80" // if the port is not specified the default port 80 is used
VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin")
VideoGrabber.RecordingInNativeFormat = true
VideoGrabber.RecordingMethod = rm_MP4
VideoGrabber.FrameGrabber = fg_Disabled
VideoGrabber.VideoRenderer = vr_None
VideoGrabber.StartRecording()
```

To close the current file and start recording to a new file while the recording is running, invoke [RecordToNewFileNow](#), this will generate a new file on the fly without losing frames.

Alternatively it is possible to enable the [RecordingTimer](#) to periodically close the current file and generate a new file.

In this case the file name is generated automatically, however the file names can be renamed on the fly from the [OnRecordingCompleted](#) event.

### RTSP, RTMP, HTTP, UDP, MMS, RTP sources and other streaming protocols

The sample code is the same, e.g.:

```
IPCameraURL = "rtsp://192.168.5.25/axis-media/media.amp?videocodec=h264&audio=1"
IPCameraURL = "udp://192.168.0.25"
IPCameraURL = "mmsh://192.168.0.25"
IPCameraURL = "rtmp://192.168.0.25/live"
IPCameraURL = "rtp://192.168.0.25"
IPCameraURL = "http://192.168.0.25/url"
etc...
```

---

Created with the Standard Edition of HelpNDoc: [Effortlessly bring your documentation online with HelpNDoc](#)

---

## Video stream of IP camera with audio of PC microphone or other audio capture device

### Video stream of IP camera with audio of PC microphone or other audio capture device

[Prev](#)

[Next](#)

Video stream of IP camera with audio of PC microphone or other audio capture device  
[Description](#)

## Capturing the video stream of an IP camera along with the audio of the soundboard microphone, or another audio capture device

This is possible starting from the version 7.1.4 of the [Datastead RTSP/RTMP/HTTP/ONVIF DirectShow Source filter](#) with the following sample code (the specific lines of code are in bold):

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.AudioDevice = VideoGrabber.AudioDeviceIndex (...) (*)
VideoGrabber.AudioSource = as_UseExternalAudio
// VideoGrabber.AudioDeviceRendering = true // if needed to render
VideoGrabber.AudioRecording = true
VideoGrabber.RecordingMethod = ...
VideoGrabber....
VideoGrabber.StartRecording()
```

(\*) replace (...) by the name of the audio device as it appears in the "audio devices" list of MainDemo.exe (the VideoGrabber.AudioDevice index is in the 0..n-1 range)

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF](#) [GetStr](#) [ONVIF](#) [SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Experience the Power and Ease of Use of HelpNDoc for CHM Help File Generation](#)

## IP cameras in MJPEG mode

### IP cameras in MJPEG mode

[Prev](#)

[Next](#)

IP Cameras in MJPEG mode

#### Description

#### IP Cameras in MJPEG mode

TVideoGrabber supports natively the IP cameras in http Motion JPEG (MJPEG) or JPG mode listed on this page, with their specific URL to use.

#### Important note:

By default the HTTP URLs in MJPEG mode are handled by TVideoGrabber, and not by the Datastead RTSP filter.

If the Datastead RTSP/RTMP/HTTP filter is installed and you want to delegate it the processing of the HTTP/MJPEG URLs, set:

```
VideoGrabber.PlayerForcedCodec = "DTSTDRTSP"
before invoking StartPreview() or StartRecording().
(to go back to the default behavior set VideoGrabber.PlayerForcedCodec="")
```

#### Starting the IP camera preview or recording

To use an IP camera as video source:

- set [VideoSource](#) = vs\_IPCamera
- set [IPCameraURL](#) = the URL of the IP camera (*the URL depends of the manufacturer*)
- invoke [SetAuthentication](#) (at\_IPCamera, ".username..", "..password...") if needed
- then invoke [StartPreview](#)() or [StartRecording](#)()

E.g.:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.IPCameraURL = "http://192.168.0.25/axis-cgi/mjpg/video.cgi";
VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin");
VideoGrabber.RecordingInNativeFormat = true
```

```
VideoGrabber.RecordingMethod = rm_MP4
VideoGrabber.AudioRecording = true
VideoGrabber.StartRecording()
```

All the TVideoGrabber properties and events related to the video capture devices apply to the IP cameras.

E.g.:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.IPCameraURL = "http://x.x.x.x/axis-cgi/mjpg/video.cgi?camera=&resolution=640x480"
VideoGrabber.StartPreview
```

Note:

- x.x.x.x must be replaced by the IP or the host name of your IP camera
- if the IP camera does not stream on the default 80 port, you have to specify the port after the IP, e.g. for the port 6015:  
http://x.x.x.x:6015/axis-cgi/mjpg/video.cgi?camera=&resolution=640x480

### Authentication

If an authentication is required to connect to the IP camera, there are 2 ways to set an username and password required:

- either by invoking [SetAuthentication](#) (at\_IPCamera, "...username...", "...password..." ) before connecting to the IP camera,
- either through the [OnAuthenticationNeeded](#) event that will occur when connecting if [SetAuthentication](#) has not been invoked yet.

### Frame rate

By default TVideoGrabber tries to stream the IP cameras at their maximal speed.  
It uses a default frame rate of 30 fps that is therefore theoretical.

You can limit the frame rate by specifying a lower value to the [FrameRate](#) property (e.g. FrameRate = 1 will receive one frame by second).

### Connection TimeOut

The connection timeout is set by default to 10000 (10 seconds), and the receive timeout to 5000 (5 seconds).

You can modify them as needed by invoking [SetIPCameraSetting](#).

### PTZ control

You can send commands to IP cameras that include PTZ control by invoking [SendIPCameraCommand](#).

E.g.:

```
SendIPCameraCommand ("http://x.x.x.x/axis-cgi/com/ptz.cgi?move=home")
SendIPCameraCommand ("http://x.x.x.x/axis-cgi/com/ptz.cgi?move=up")
tc...
```

### List of the IP camera URL formats to use, depending of the manufacturer

*This list is not exhaustive. If you don't know the URL to use for your IP camera, feel free to contact us at [support@datastead.com](mailto:support@datastead.com).*

The URLs below must begin by http://ip:port or http://hostname:port

Examples for the axis camera below, with the following URL format: **/axis-cgi/mjpg/video.cgi**.

1) let's suppose the camera IP is 212.45.2.14, the URL to use is:

e.g. on the default port 80: `http://212.45.2.14/axis-cgi/mjpg/video.cgi`

e.g. if setup on the port 10120: `http://212.45.2.14:10120/axis-cgi/mjpg/video.cgi`

2) let's suppose the camera host name is myipcamera.net, the URL to use is:

e.g. on the default port 80: `http://myipcamera.net/axis-cgi/mjpg/video.cgi`

e.g. if setup on the port 9580: `http://myipcamera.net:9580/axis-cgi/mjpg/video.cgi`

### **A-Linking**

`/GetData.cgi`

Airlink

`/mjpeg.cgi`

`/cgi/mjpg/mjpeg.cgi`

`/cgi/jpg/image.cgi`

### **Airlive**

`/video.mjpg`

`/mjpg/video.mjpg`

### **Airwave**

`/cgi-bin/pusher.cgi`

### **Arecont**

`/mjpeg?res=full&x0=0&y0=0&x1=100%&y1=100%&quality=12&doublescan=0&fps=1&ver=HTTP/1.1`

`/image?res=half&x0=0&y0=0&x1=1600&y1=1200&quality=15&doublescan=0`

### **Aviosys**

`/GetData.cgi`

### **Aviosys 9060-I**

`/cgi-bin/Stream?Video?Authorization=`

### **Axis IP camera in MJPEG format**

`/axis-cgi/mjpg/video.cgi`

`/axis-cgi/mjpg/video.cgi?resolution=352x240` *(the resolution specified must be available on the IP camera, look at the camera video settings panel)*

### **Axis (IP video server, several cameras) in MJPEG format**

`/axis-cgi/mjpg/video.cgi?camera=1&resolution=320x240`

`/axis-cgi/mjpg/video.cgi?camera=2&resolution=320x240`

`/axis-cgi/mjpg/video.cgi?camera=3&resolution=320x240`

`/axis-cgi/mjpg/video.cgi?camera=4&resolution=320x240`

`/axis-cgi/mjpg/video.cgi?camera=quad&resolution=704x576` (4 x 4 layout of the 4 inputs)

### **Bowya ([here](#))**

`/video.cgi`

### **Bosch**

`/snap.jpg`

### **Canon (VB-C50, VB-C60, etc...)**

`/-vvhttp-01-/`

`/-vvhttp-01-/GetOneShot`

`/-vvhttp-01-/GetOneShot?frame_count=no_limit`

`/-vvhttp-01-/GetStillImage`

### **Convision**

`/fullsize.push?camera=1&sleep=15`

**Digicom**

/mjpeg.cgi

**D-Link (DSC2121)**

/video/mjpg.cgi

**D-Link**

/video/mjpg.cgi

/video.cgi

/mjpeg.cgi

/cgi-bin/video.jpg

/IMAGE.jpg

/cgi-bin/video.vam

/\_gCvImage.jpg

**EasyN**

/video.cgi

/videostream.cgi

/videostream.cgi?resolution=8

/videostream.cgi?resolution=8&rate=13

/videostream.cgi?user=username&pwd=password

**Edimax**

/jpg/image.jpg

/mjpg/video.mjpg

/snapshot.cgi

**Ego PT-200**

/cgi-bin/sf.cgi

**Foscam**

/videostream.cgi

/snapshot.cgi

**Fulicom FC-CR1060**

/cgi-bin/sf.cgi

**Gadspot**

/Jpeg/CamImg.jpg

/GetData.cgi?Status=0

**Goscam**

/cgi-bin/Stream?Video?Acc=**USER**?Pwd=**PASSWORD**?webcamPWD=RootCookies00000

*(replace USER and PASSWORD by their respective values)*

**Kingnow PT200**

/cgi-bin/sf.cgi

**Hamlet**

[http://www.hamletcom.com/ProductDetails.aspx?ProductCode\\_EQ\\_HNIPC30W](http://www.hamletcom.com/ProductDetails.aspx?ProductCode_EQ_HNIPC30W)

/mjpeg.cgi

**Intellinet**

/jpg/image.jpg

or

/temp/image.jpg

*(you may have to enable the direct HTTP image access in the setup of the camera)*

**Intellinet NCS18**

/jpg/image.jpg

**IP Cam waterproof Infrared IP Camera:** <http://ipcam.en.ecplaza.net/11.asp>

/snapshot.cgi

### **IQeye**

/now.jpg?snap=spush

### **JVC** (e.g. VN-X35U/235U)

/api/video?encode=jpeg&framerate=15&boundary=on

### **Linksys**

/img/snapshot.cgi?size=2

/img/video.mjpeg

/img/mjpeg.cgi

(MJPEG mode activated with SendIPCameraCommand ("/adm/file.cgi?h\_videotype=mjpeg&todo=save"))

mms://x.x.x.x/img/video.asf

(ASF mode activated with SendIPCameraCommand ("/adm/file.cgi?h\_videotype=mpeg4&todo=save"))

### **Linudix**

/cgi-bin/nph-update\_4ch.cgi?ch=1

### **Lumenera**

/cgi-bin/nph-video

### **Marmitek**

/cgi/mjpg/mjpeg.cgi

#### *Predefined positions:*

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?

action=move&Cmd=Position1');

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?

action=move&Cmd=Position2');

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?

action=move&Cmd=home');

...

#### *Pan/tilt control:*

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?action=move&Cmd=up');

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?action=move&Cmd=right');

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?action=move&Cmd=down');

videograbber1.SendIPCameraCommand('http://[IPaddress]/cgi/admin/ptctrl.cgi?action=move&Cmd=left');

### **Mobotix**

/record/current.jpg

/control/faststream.jpg?stream=full

/faststream.jpg?stream=full&fps=1.0 (1 fps)

/faststream.jpg?stream=full&fps=3.0 (1 fps)

/faststream.jpg?stream=full&fps=0 (max frame rate)

### **Moxa**

//cgi-bin/video.jpg

### **Panasonic**

/nphMotionJpeg?Resolution=640x480&Quality=Clarity

/cgi-bin/nphContinuousServerPush

/SnapshotJPEG?mode=Refresh

/cgi-bin/camera

### **Pixord**

/Getimage.cgi

/Getimage?camera=1&fmt=full (full size)

/Getimage?camera=1&fmt=qsif (half size)

/Getimage?camera=1&fmt=sif (quarter size)

### **PLANET ICA-108**

/jpg/image.jpg

**Qnap**

/cgi/mjpg/mjpeg.cgi

**Samsung SNB**

/video?submenu=mjpg

/video?submenu=jpg

**Sanyo**

/liveimg.cgi?serverpush=1 (*MJPEG mode*)

/liveimg.cgi (*Jpeg mode*)

**Sharkx**

/stream.jpg

**Silicon**

/snapshot.cgi

**Skyway Security**

/GetData.cgi?Status=0

/Jpeg/CamImg.jpg

**Sony**

/image

/image?speed=0

/onshotimage.jpg

**Shenzen Sunsky S-NC-0201:** <http://www.sun-usb.com/product/details/S-NC-0201.htm>

/cgi-bin/sf.cgi

**Surecom**

/mjpeg.cgi

**Swann IP-3G ConnectCam 1000**

/cgi/jpg/image.cgi

**Topcom**

[http://www.topcom.net/fiche.asp?p\\_EQ\\_wirelessipcam2000](http://www.topcom.net/fiche.asp?p_EQ_wirelessipcam2000)

/mjpeg.cgi

**Toshiba**

/\_\_live.jpg?&&&

getstream.cgi?10&10&&&10&0&0&0&0

**TP-Link**

/jpg/image.jpg

/video.mjpg

(the "/video.mjpg" MJPEG streaming is available only after disabling the RTSP streaming and rebooting the camera)

**Trendnet:** <http://trendnet.com/products/products.asp?cat=48>

/goform/video (e.g. TV-IP201W)

/goform/video2 (e.g. TV-IP201W)

/cgi/mjpg/mjpg.cgi (e.g. TV-IP410)

/GetData.cgi

/image.jpg

**Vilar**

/cgi-bin/sf.cgi

**Vivotek**

/video.mjpg (*IP71.., IP 73.., VS71.. cameras. Be sure to enable the MJPG mode in the video settings*)



/cgi-bin/video.jpg  
/cgi-bin/viewer/video.jpg

#### Y-Cam

/stream.jpg

#### Zavio

/jpg/image.jpg

#### See Also

[Opening a clip or an IP URL from a background thread without blocking the main thread](#) [Player features](#) [TOnThreadSync](#) [TThreadSyncPoint](#) [EnableThreadMode](#) [OnThreadSync](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## IR Cut Filter of Axis cameras

### IR Cut Filter of Axis cameras

[Prev](#)

[Next](#)

IR Cut Filter of Axis cameras

#### Description

It is possible to control the IR Cut Filter of IP cameras through the [Datastead RTSP/RTMP/HTTP/ONVIF DirectShow Source Filter](#) v7.2.2.1 or newer, that must be installed first.

**First start the preview of the Axis IP camera with the following sample code, e.g.:**

```
VideoGrabber.VideoSource = vs_IPCamera
```

```
VideoGrabber.IPCameraURL = "rtsp://192.168.5.29/axis-media/media.amp?videocodec=h264&audio=1";
```

```
VideoGrabber.SetAuthentication (at_IPCamera, "root", "password");
```

then, once the preview is running:

- to set the IR Cut Filter state, invoke one of the following commands:

```
VideoGrabber.ONVIF_SetStr ("RTSP_Source_Axis_IrCutFilter_str", "enabled")
```

or

```
VideoGrabber.ONVIF_SetStr ("RTSP_Source_Axis_IrCutFilter_str", "disabled")
```

or

```
VideoGrabber.ONVIF_SetStr ("RTSP_Source_Axis_IrCutFilter_str", "auto")
```

**- to retrieve the IR Cut Filter state, invoke:**

```
function VideoGrabber.ONVIF_GetStr ("RTSP_Source_Axis_IrCutFilter_str", Value) : Boolean
```

Value returns one of the following values: "enabled", "disabled", or "auto"

#### See Also

[TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## Asynchronous vs Asynchronous connection

### Asynchronous vs Asynchronous connection

[Prev](#)

[Next](#)

Synchronous vs Asynchronous connection

## Description

TVideoGrabber can connect to the IP cameras and URLs:

### - synchronously if OpenURLAsync is disabled

In this case when invoking StartPreview() or StartRecording(), the function blocks until the connection succeeds or fails.

### - synchronously if OpenURLAsync is enabled (default)

In this case, when invoking StartPreview() or StartRecording(), the function returns immediately so the main thread is not blocked and the application continues to respond to key and mouse events.

Once the connection is completed, the OnPreviewStarted() or OnRecordingStarted() event occurs to notify the application that the component is running.

If the connection fails, the [OnLog](#) event occurs with a "e\_failed\_to\_start\_preview" or "e\_failed\_to\_start\_recording" [LogType](#) error.

Example of generic StartPreview() / StartRecording() set of functions that handles the asynchronous connection result through the OnPreviewStarted() / OnRecordingStarted() and OnLog() events:

```
function StartPreview()

    VideoGrabber.OpenURLAsync = true // default
    VideoGrabber.VideoSource = vs_IPCamera
    VideoGrabber.IPCameraURL =
rtsp://192.168.0.25/axis-media/media.amp?videocodec=h264

    VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin");
    VideoGrabber.StartPreview()

function StartRecord()

    VideoGrabber.OpenURLAsync = true // default
    VideoGrabber.VideoSource = vs_IPCamera
    VideoGrabber.IPCameraURL =
rtsp://192.168.0.25/axis-media/media.amp?videocodec=h264&audio=1

    VideoGrabber.SetAuthentication (at_IPCamera, "root", "admin");
    VideoGrabber.RecordingInNativeFormat = true
    VideoGrabber.RecordingMethod = rm_MP4
    VideoGrabber.AudioRecording = true
    VideoGrabber.StartRecording()

event function OnRecordingStartedEvent(Sender, Filename)

    ShowMessage (filename + " recording started!")

event function OnPreviewStartedEvent(Sender)

    ShowMessage ("preview started!")

event function OnLogEvent(Sender, Logtype, Severity, Errormsg)

    if (LogType = e_failed_to_start_recording)
```

```
ShowMessage ("error: recording failed to start")

else if (LogType = e_failed_to_start_preview)
    ShowMessage ("error: preview failed to start")
```

---

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## Auto-reconnection

### Auto-reconnection

[Prev](#)

[Next](#)

Auto-Reconnection

#### Description

##### Auto-Reconnection of an URL

- 1) If the IP camera auto-reconnection is enabled (default setting):
  - when the connection is lost, the [OnDeviceReconnecting](#) event occurs and the RTSP filter tries to reconnect to the IP camera
  - when the camera is connected again, the [OnDeviceReconnected](#) event occurs
- 2) If the IP camera autoreconnection is disabled:

- when the connection is lost, the [OnDeviceLost](#) event occurs and TVideoGrabber stops.

To disable the auto-reconnection, add:

>autoreconnect=0

at the end of the URL, e.g.:

VideoGrabber.IPCameraURL = "rtsp://192.168.5.1/live.sdp>autoreconnect=0"

---

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

---

## NTP time of IP cameras

### NTP time of IP cameras

[Prev](#)

[Next](#)

NTP time of IP cameras

#### Description

If the camera streams expose the NTP time, the NTP time of the current frame being displayed can be obtained as 100ns units:

- at any time by invoking `int64 NTPTime = VideoGrabber.GetFrameInfo (0, fi_NTPFrameTime)`
- though the [TFrameInfo](#) structure pointer of the [OnFrameBitmap](#) event: `FrameInfo->NTPFrameTime`

Note :

- if the NTP server is configured on the IP camera, the typical time value should look like the [current Unix epoch time](#).
- if the NTP server is not configured the value may start from a given value, but **the NTP time of the current frame minus the NTP time of the first frame received** can be anyway used to calculate the current stream duration since the preview started.

---

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

---

## Using the TVideoGrabber http/mjpeg decoder when the Datastead RTSP filter is installed

### Using the TVideoGrabber http/mjpeg decoder when the Datastead RTSP filter is installed

[Prev](#)

[Next](#)

Using the TVideoGrabber http/mjpeg decoder when the Datastead RTSP filter is installed

## Description

When the [Dastead RTSP/RTMP/HTTP/ONVIF DirectShow Source Filter](#) is installed, the capture/decoding of HTTP streams is delegated to it.

To force TVideoGrabber to use its embedded HTTP/MJPEG decoder anyway, set:

`VideoGrabber.PlayerForcedCodec := "HTTP_TVG";`

before invoking `StartPreview()` or `StartRecording()`

(This can be cancelled by setting `VideoGrabber.PlayerForcedCodec := ""`)

---

Created with the Standard Edition of HelpNDoc: [Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

---

## Audio

---

Created with the Standard Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

---

### Audio capture devices

#### Audio capture devices

[Prev](#)
[Next](#)

Audio capture devices overview.

#### Description

##### Audio Source

If [AudioSource](#) = **as\_Default** (default value) the audio output of the video capture device will be used for recording (if this device exposes an audio output, e.g. like a DV camcorder), otherwise the audio output of the current audio capture device will be used.

If [AudioSource](#) is set to **as\_UseExternalAudio**, the current audio capture device will be used for recording, even if the video capture device has an audio out.

##### Audio capture device

The current audio capture device is selected by [AudioDevice](#), which is an index in the [AudioDevices](#) list, that contains [AudioDevicesCount](#) items.

It is possible to select the audiop capture device programmatically by name by using the [FindIndexInListByName](#) function, e.g.:

The name of the current audio capture device is reported by [AudioDeviceName](#).

```
VideoGrabber.AudioDevice := VideoGrabber.FindIndexInListByName (VideoGrabber.AudioDevices
```

*Note: [AudioDevices](#) and `AudioDevices` count are also available as global variables in Delphi and C++Builder.*

##### Device-dependent properties

When an audio capture device is selected, its [device-dependent properties](#) are reloaded from the registry and the [OnAudioDeviceSelected](#) event occurs, therefore any control that uses e.g. the audio inputs should be refreshed from this event.

##### Audio inputs

After selecting an audio capture device, the [AudioInputs](#) lists returns the list of the audio inputs available for this audio capture device. An audio input can be selected by assigning the index of audio input in the list to the [AudioInput](#) property (in the 0..[AudioInputsCount](#) - 1 range).

##### Audio input level

The audio input level can be adjusted with [AudioInputLevel](#) (in the 0..65535) range.

##### Audio input balance

The audio input balance can be adjusted with [AudioInputBalance](#) in the -32768...32767 range (the center point is 0).

The audio input balance can be available or not, depending of the audio input. When preview or recording is running, [IsAudioInputBalanceAvailable](#) returns ts\_True if the audio balance is available, and ts\_False if not. If preview and recording are stopped, it returns ts\_Undefined.

#### Audio recording

The recording of the current audio capture device is activated when [AudioRecording](#) is enabled.

#### Audio rendering

The audio rendering is activated when [AudioDeviceRendering](#) is enabled.

The audio volume can be adjusted with [AudioVolume](#) and the audio balance with [AudioBalance](#).

To mute the audio volume enable/disable the [MuteAudioRendering](#) property.

These settings affect only the volume and balance of the audio rendering, not the volume and balance of the audio recording.

#### Device connection/disconnection

When an audio capture device is connected or disconnected, the [OnDeviceArrivalOrRemoval](#) event occurs. This event reports the name and index of the audio capture device in the global [AudioDevices](#) list.

After an audio capture device has been disconnected, [IsAudioDeviceConnected](#) returns true, until the device is reconnected.

If the related device is used by preview or recording, the [OnDeviceLost](#) event occurs when the device is disconnected, and the preview or recording stops.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

---

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

---

## Audio rendering

### Audio rendering

[Prev](#)

[Next](#)

Audio rendering

#### Description

#### Enabling the audio rendering

The audio rendering is delivered when [AudioDeviceRendering](#) = true, excepted for the player for which the audio is rendered when [PlayerAudioRendering](#) = true

#### Volume and balance

The volume is usually adjusted with [AudioVolume](#), and the balance with [AudioBalance](#).

These settings does not interact with the default sound card, unless the default sound card has been selected as [AudioDevice](#) in the [AudioDevices](#) list.

It is possible to modify the global audio volume and balance of the default sound card with [SpeakerVolume](#) and [SpeakerBalance](#).

#### Choosing the audio renderer

By default, the default DirectSound Audio Renderer is used ([AudioRenderer](#) = -1).

To select a different audio renderer, simply assign to [AudioRenderer](#) the index of the audio renderer to use, chosen in the [AudioRenderers](#) list.

### Selecting the right or left channel

It is possible to convert the right channel or the left channel as "mono" and to mute either the left, right or both channels, or convert a mono channel in stereo with [AudioChannelRenderMode](#).

### Selecting the audio stream

When the clip has multiple audio streams it is possible to select one audio stream, see [AudioStreamNumber](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## Audio levels and VU-meters

### Audio levels and VU-meters

[Prev](#)

[Next](#)

Audio levels and VU-meters

#### Description

#### Audio levels

The audio levels are returned by the [OnAudioPeak](#) event (type [TOnAudioPeak](#)) when the [AudioPeakEvent](#) property is enabled.

The event returns the peak values of the left and right channels in percentage or in DB.

#### VU-Meters

VU-Meters are integrated in the component (analog or bargraph) and can be easily displayed.

Proceed as follows:

1. put 2 panel (or image) controls on the form, one for the left channel, one for the right channel. Let's suppose they are named Panel1 and Panel2.

2. set VUMeter = vu\_Analog (or vu\_Bargraph)

3. assign the handle of the panel (or the image) controls as follows:

```
VideoGrabber.SetVUMeterSetting(VU_LEFT, vu_Handle, Panel1.Handle)
VideoGrabber.SetVUMeterSetting(VU_RIGHT, vu_Handle, Panel2.Handle)
```

4. you can adjust the other VU-Meter settings (mainly colors), if needed, by invoking [SetVUMeterSetting](#).

#### Note:

to detach the VU-Meters from their panel (or image) controls, invoke:

```
VideoGrabber.SetVUMeterSetting(VU_LEFT, vu_Handle, 0)
VideoGrabber.SetVUMeterSetting(VU_RIGHT, vu_Handle, 0)
```

#### VU-meters and bargraphs overlayed over the video frames

Up to 6 vu-meters/bargraphs are availables (from index 0 to 5).

(for backward compatibility the VUMeter property is kept but it activates only the VUMeters 0 and 1)

Additionally, now the VU-Meters and bargraphs can display custom values updated in real-time, the custom values must be in the 0..100 range.

To activate any of the 6 VU-meters, instead of setting the VUMeter property, invoke [SetVUMeter\\_Enabled](#).

E.g. to activate a bargraph, invoke `SetVUMeter_Enabled(index, vu_BargraphOverlay)`

So is possible to use;

- the 2 originals VU-meters (index 0 and 1)
- 2 additional VU-meters or bargraphs (index 2 and 3) that can display audio or custom values
- 2 additional VU-meters or bargraphs (index 4 and 5) that can display custom values ONLY

Example to activate the bargraphs 2 and 3:

```
VideoGrabber.SetVUMeter_Enabled(2, vu_BargraphOverlay);
VideoGrabber.SetVUMeterSetting(2, vu_Transparent, 0);
VideoGrabber.SetVUMeterSetting(2, vu_OverlayLeft, 10);
VideoGrabber.SetVUMeterSetting(2, vu_OverlayTop, 10);
VideoGrabber.SetVUMeterSetting(2, vu_OverlayWidth, 100);
VideoGrabber.SetVUMeterSetting(2, vu_OverlayHeight, 60);
VideoGrabber.SetVUMeter_Enabled(3, vu_BargraphOverlay);
VideoGrabber.SetVUMeterSetting(3, vu_Transparent, 0);
VideoGrabber.SetVUMeterSetting(3, vu_OverlayLeft, 200);
VideoGrabber.SetVUMeterSetting(3, vu_OverlayTop, 500);
VideoGrabber.SetVUMeterSetting(3, vu_OverlayWidth, 100);
VideoGrabber.SetVUMeterSetting(3, vu_OverlayHeight, 60);
```

and to use the bargraph 3 as custom bargraph:

```
VideoGrabber.SetVUMeterSetting(3, vu_CustomPercentValue, 40); // e.g. for 40%
```

**Note: to customize the VU-meter or bargraph foreground and background color**, pass the color as RGB hexadecimal value, e.g.:

Delphi:

```
SetVUMeterSetting(2, VidGrab.TVUMeterSetting.vu_NormalColor, $FFFF00)
```

C#:

```
SetVUMeterSetting(2, VidGrab.TVUMeterSetting.vu_NormalColor, 0xFFFF00)
```

VB:

```
SetVUMeterSetting(2, VidGrab.TVUMeterSetting.vu_NormalColor, CType(&Hffff00, IntPtr))
```

## See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

## Recording

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

### Recording methods and properties

#### Recording methods and properties

[Prev](#)

[Next](#)

Recording overview.

#### Description

#### Recording methods and properties

##### Video and audio capture devices

- if [AudioRecording](#) is disabled, only the [VideoDevice](#) video source (chosen in the [VideoDevices](#) list) will be recording to the file.
- if [AudioRecording](#) is enabled, the current [VideoDevice](#) and the current [AudioDevice](#) (chosen in the [AudioDevices](#) list) will be captured.
- if the audio and video capture devices are different and the audio of the video clip recorded is not in sync with the video, enable [AudioSyncAdjustmentEnabled](#) and adjust the number of audio samples to delay with [AudioSyncAdjustment](#).

#### Video format

The video format is determined by the extension of the [RecordingFileName](#)

E.g. to record a MP4 video set .mp4 as extension, to record a Matroska video set .mkv as extension.

If the file name is generated automatically, specify the video format with the RecordingMethod, e.g.

[RecordingMethod](#) = rm\_MP4 will record a ".mp4" video clip.

RECORDING BY USING THE [DATASTEAD ENCODER](#) (recommended)

See [Recording through the Datastead Encoder](#)



## RECORDING BY USING THIRD-PARTY CODECS

- select the third-party codec by its name, e.g.:
- set [VideoCompressor](#) = [VideoCompressorIndex](#) ("x264vfw - H.264/MPEG-4 AVC codec")
- set [CompressionMode](#) = cm\_CompressOnTheFly

Starting the recording immediately

- once the settings above applied, invoke [StartRecording\(\)](#)

Starting the recording preview but not writing immediately to the file

- once the settings above applied:

- set [HoldRecording](#) = true

- invoke [StartRecording\(\)](#)

this starts the video preview and recording but does not write yet to the file.

- later, to begin writing to the file, invoke [ResumeRecording\(\)](#)

## Stopping writing to the current file and starting writing to a new file while the recording is running

Invoke [RecordToNewFileNow](#)

## Stopping the recording

- invoke [Stop\(\)](#)

## Playing a video currently being recorded

This is possible when recording in .MP4 or .MKV format by enabling the [VideoPlayableWhileRecording](#) property.

## Delayed start/stop

The [RecordingTimer](#) property lets you start, stop or restart periodically the recording after an interval of time:

[RecordingTimer](#) = rt\_Disabled : normal start/stop

[RecordingTimer](#) = rt\_RecordToNewFile : a new recording file is generated each [RecordingTimerInterval](#) seconds

[RecordingTimer](#) = rt\_StopRecording : the recording will stop after [RecordingTimerInterval](#) seconds

[RecordingTimer](#) = rt\_StartRecording : the recording starts in preview mode only, the real recording will begin after [RecordingTimerInterval](#) seconds

## Creating a new video clip on the fly

While the recording is running, invoke [RecordToNewFileNow](#) (*NewFileName*, true) and a new video clip will be created.

- either pass the new file name to be created, either pass an empty string to let TVideoGrabber generate the new file name automatically, according to the [AutoFileName](#) settings.

To create small video clips having the same duration:

- set [RecordingTimerInterval](#) to the desired period (expressed in seconds)
- set [RecordingTimer](#) = rt\_RecordToNewFileNow

*Note that this feature is **not compatible** with:*

- the recompression after recording (be sure that you use [CompressionMode](#) = cm\_NoCompression or cm\_CompressOnTheFly),
- the use of a preallocated recording file (be sure that [PreallocCapfileEnabled](#) = false)

## Specifying a maximum file size

To limit the recording file size, specify a [RecordingFileSizeMaxInMB](#) value > 0.

During the recording, each time the size specified is exceeded, a new file is generated on the fly, according to the auto file name settings.

## Holding the beginning of the recording for an accurate recording start time

By default, when invoking [StartRecording](#) the recording starts immediately, but building the recording graph requires a few seconds, so the moment when the recording starts after invoking [StartRecording](#) is never accurate.

To avoid this problem, enable the [HoldRecording](#) property before invoking [StartRecording](#).

When [HoldRecording](#) is enabled, the recording graph is built, and then the [OnRecordingReadyToStart](#) event occurs.

When this event occurs, you can invoke [ResumeRecording](#) to really start the recording, or [StopRecording](#) to cancel it.

E.g.:

...



```

procedure TForm1.Button1Click (Sender: TObject);
begin
VideoGrabber1.HoldRecording := True;
VideoGrabber1.StartRecording;
end;
...
procedure TForm1.VideoGrabberRecordingReadyToStart(Sender: TObject);
begin
if ConditionToStartRecording then begin
VideoGrabber.ResumeRecording;
end
else begin
VideoGrabber.StopRecording;
end;
end;
end;

```

### Recording file name

The recording file name can be chosen by assigning the [RecordingFileName](#) property before invoking [StartRecording](#).

If [RecordingFileName](#) is left blank, a file name is generated automatically according to the [AutoFileName](#), [AutoFileNameDateTimeFormat](#), [AutoFileNameMinDigits](#), [AutoFilePrefix](#) and [CaptureFileExt](#) properties.

*By example if you customize the file name as follows:*

*VideoGrabber.AutoFileName = fn\_DateTime;*

*VideoGrabber.AutoFileNameDateTimeFormat = "dd-mm-yy\_hh-mm-ss"*

*VideoGrabber.AutoFilePrefix = "file\_"*

*the recording file name will be file\_13-02-13\_15-27-45.avi*

As soon as recording begins, this file name can be retrieved with [Last\\_Recording\\_FileName](#).

When the recording ends, this file name is returned by the [OnRecordingCompleted](#) event.

### Current file size of the video clip beeing recorded

The [RecordingKBytesWrittenToDisk](#) reports the current (growing) file size in KB while the recording is running.

Invoke this function periodically (e.g. every 10 seconds) to retrieve on the fly the current file size of the video clip being recorded.

When the recording ends, invoke [RecordingKBytesWrittenToDisk](#) from the [OnRecordingCompleted](#) event to get the final file size of the recorded clip.

You can find the corresponding sample code in the OnFrameProgress event of the MainDemo project.

### Pausing / resuming the AVI recording

See [AVI pause/resume during recording](#)

### Events

- if [HoldRecording](#) is enabled, when invoking StartRecording the [OnRecordingReadyToStart](#) event occurs, and then you can invoke [ResumeRecording](#) to really start to record.
- when the recording really begins, the [OnRecordingStarted](#) event occurs,
- when the recording ends, the [OnRecordingCompleted](#) event occurs.

### Frame capture

It is possible to capture frames during recording when the [frame grabber](#) is enabled. See the [Frame Capture](#) chapter.

### Frame overlay

It is possible to perform graphics or text overlay during recording when the [frame grabber](#) is enabled. See the [Frame Capture](#) chapter.

### Quality of the video window during Recording

If the quality of the preview is not critical during recording, we recommend to set [VideoRenderer](#) = [vr\\_RecordingPriority](#) before invoking StartRecording to assign the resources prioritarily to the recording.

As the CPU load is critical during recording, in certain cases, preventing the video frames to be stretched to the video window size may save CPU, especially if the video frame is resized to a size larger than the original size of the video frame.

To prevent the video frames to be resized:

- if you want the control to be resized automatically, set [AdjustPixelAspectRatio](#) = false **and** [Display\\_AutoSize](#) = true
- if you want to specify yourself the size of the control, set [AdjustPixelAspectRatio](#) = false **and** [Display\\_AutoSize](#) = false **and** [Display\\_AspectRatio](#) = ar\_NoResize

### Back-timed recording (preroll)

It is possible to start recording with a preroll a few seconds before an event occurs.

E.g. when motion detection is used, this allows to capture the full video sequence by starting just before the motion detection event occurs.

See the [Back-timed recording \(preroll\)](#) chapter.

### CPU and disk requirements

In order to minimize dropped frames, capturing to AVI in real time requires high CPU and disk availability, therefore:

- prefer fast disks / fast CPU platforms,
- prefer Windows 2000 / Windows XP platforms rather than Win98 or WinME platforms,
- prefer NTFS partitions rather than FAT32 partitions.

If several video clips are recorded at the same time and lot of dropped frames occur, we recommend to use SCSI disks for better disk performances.

### Dropped frames

As video capture is synchrone, when the time required to record a given video frame becomes longer than the time between 2 video frames, frames are dropped.

An excessive amount of dropped frames cause jerky AVI playback and bad audio/video synchronization.

When dropped frames problems occur, act on the following points:

- check that [StoragePath](#) saves to a fast disk (avoid networked drives),
- disable the frame grabber on the AVI stream (set [FrameGrabber](#) = fg\_PreviewStream), or disable it totally (set [FrameGrabber](#) = fg\_Disabled)
- reduce the video size,
- reduce the frame rate ([FrameRate](#) property or [DVReduceFrameRate](#) for DV sources).

### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TCompressionType](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TOnVideoCompressionSettings](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [CompressionMode](#) [CompressionType](#) [Encoder](#) [GetInt](#) [Encoder](#) [Pause](#) [Encoder](#) [Resume](#) [Encoder](#) [SetInt](#) [Encoder](#) [SetStr](#) [Encoders](#) [CreateInstanceForRecording](#) [Encoders](#) [CreateInstanceForStreaming](#) [Encoders](#) [RemoveAllInstances](#) [Encoders](#) [RemoveInstance](#) [GetVideoCompressionSettings](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [RefreshDevicesAndCompressorsLists](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#) [VideoPlayableWhileRecording](#)

## Recording through the Datastead Encoder

### Recording through the Datastead Encoder

[Prev](#)

[Next](#)

Recording through the Datastead Encoder

#### Description

The installation and functions overview are described in the [Datastead Encoder](#) chapter.

It is possible:

1. either to record through the Datastead recording functions, allowing to use TVideoGrabber's OnRecording... events, in this case TVideoGrabber is started by invoking **StartRecording()**
2. either to record through an external Encoder instance (not managed by TVideoGrabber), in this case TVideoGrabber is started by invoking **StartPreview()**, that will send the audio/video samples to all the active Encoder instances.

SAMPLE CODE

**1. sample code for a recording using the TVideoGrabber recording functions:**  
managed by TVideoGrabber, that controls it and raises the corresponding recording events

#### IMPORTANT NOTE:

The TVideoGrabber recording features (RecordingFileName / CompressionMode / StartRecording / PauseRecording / ResumeRecording / RecordToNewFileNow, etc...) invoke the internal Encoder instance with ID 0.

In this case, **do not invoke Encoders\_CreateInstanceForRecording to pass the recording file name instead of setting RecordingFileName**, this is already controlled internally by TVideoGrabber.

Only Encoder\_SetInt and Encoder\_SetStr should be invoked with ID 0 to configure the parameters, as shown in the MainDemo project -> "multipurpose" tab.

If you prefer to configure and control the recording yourself by configuring a recorder instance with Encoders\_CreateInstanceForRecording, run TVideoGrabber in preview mode only (by invoking StartPreview), it will pass the audio/video samples to the instance you have created, as described in the paragraph 2. below.

#### A) STARTING THE RECORDING

```
#define ENCODER_RECORDING_ID 0 // the encoder with ID 0 is the TVideoGrabber's
embedded encoder, this will never change
// in the MainDemo project, "video source" tab
VideoGrabber.VideoDevice = VideoGrabber.VideoDeviceIndex ("c922 Pro Stream
Webcam");
VideoGrabber.AudioDevice = VideoGrabber.AudioDeviceIndex ("Microphone (C922
Pro Stream Webcam)");
VideoGrabber.VideoCompressor = VideoCompressorIndex ("Datastead Multipurpose
Encoder");
VideoGrabber.AudioCompressor = AudioCompressorIndex ("Datastead Multipurpose
Encoder");
// in the MainDemo project, "multipurpose" tab
VideoGrabber.Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_Bitrate_kb,
2000);
VideoGrabber.Encoder_SetStr (ENCODER_RECORDING_ID, Enc_Video_Codec, "hevc");
VideoGrabber.Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_Thread_Count, 4);
// default 1
// VideoGrabber.Encoder_SetInt (ENCODER_RECORDING_ID, Enc_Video_GPU_Encoder,
integer (Enc_GPU_NVidia_NVENC)); to enable the GPU encoding through NVidia (to
select any GPU available set Enc_GPU_Auto)
VideoGrabber.Encoder_SetStr (ENCODER_RECORDING_ID, Enc_Audio_Codec, "aac");
// in the MainDemo project, "recording" tab
VideoGrabber.RecordingMethod = rm_MP4; // to generate the file name
automatically,
// VideoGrabber.RecordingFileName = "VideoClip.mp4"; // if the file name is
```

specified, it determines the recording method, no need to set RecordingMethod above

```
VideoGrabber.AudioRecording = true;
VideoGrabber.CompressionMode = cm_CompressOnTheFly;
VideoGrabber.StartRecording();
```

#### B) PAUSING THE RECORDING

```
VideoGrabber.PauseRecording();
```

#### C) RESUMING THE RECORDING

```
VideoGrabber.ResumeRecording();
```

#### D) RECORDING TO A NEW FILE

```
VideoGrabber.RecordToNewFileNow ("c:/folder/newvideoclip.mp4");
```

#### E) STOPPING THE RECORDING

```
VideoGrabber.Stop();
```

### 2. same sample code to recording using an external encoder instance

not managed by TVideoGrabber, that just sends the audio/video samples to the encoder(s)

#### A) STARTING THE RECORDING

```
int m_EncoderInstanceID = -1;
// in the MainDemo project, "video source" tab
VideoGrabber.VideoDevice = VideoGrabber.VideoDeviceIndex ("C922 Pro Stream Webcam");
VideoGrabber.AudioDevice = VideoGrabber.AudioDeviceIndex ("Microphone (C922 Pro Stream Webcam)");
m_EncoderInstanceID = VideoGrabber.Encoders_CreateInstanceForRecording ("VideoClip.mp4");
if (m_EncoderInstanceID > -1)
VideoGrabber.Encoder_SetInt (m_EncoderInstanceID, Enc_Video_Enabled_bool, 1);
VideoGrabber.Encoder_SetInt (m_EncoderInstanceID, Enc_Video_Bitrate_kb, 2000);
VideoGrabber.Encoder_SetStr (m_EncoderInstanceID, Enc_Video_Codec, "hevc");
VideoGrabber.Encoder_SetInt (m_EncoderInstanceID, Enc_Video_Thread_Count, 4); // default 1
// VideoGrabber.Encoder_SetInt (m_EncoderInstanceID, Enc_Video_GPU_Encoder, integer (Enc_GPU_NVidia_NVENC)); to enable the GPU encoding through NVidia (to select any GPU available set Enc_GPU_Auto)
VideoGrabber.Encoder_SetInt (m_EncoderInstanceID, Enc_Audio_Enabled_bool, 1);
VideoGrabber.Encoder_SetStr (m_EncoderInstanceID, Enc_Audio_Codec, "aac");
VideoGrabber.Encoder_SetInt (m_EncoderInstanceID, Enc_Audio_Bitrate_kb, 64);
VideoGrabber.AudioDeviceRendering = true;
VideoGrabber.StartPreview();
```

#### B) PAUSING THE RECORDING

```
VideoGrabber.Encoder_Pause (m_EncoderInstanceID);
```

#### C) RESUMING THE RECORDING

```
VideoGrabber.Encoder_Resume (m_EncoderInstanceID);
```

#### D) RECORDING TO A NEW FILE

bool OpenInPausedState = false; // to record to the new file immediately, otherwise the current file is closed and the recording to the next file will start when invoking Encoder\_Resume

```
VideoGrabber.Encoder_NewOutputFile(m_EncoderInstanceID, "c:/folder/newvideoclip.mp4", OpenInPausedState);
```

#### D) STOPPING THE RECORDING ONLY

```
VideoGrabber.Encoder_NewOutputFile(m_EncoderInstanceID, "c:/folder/AnyNewFileName.mp4", true);
```

// this closes the current file and prepare for the next file. If you never invoke Encoder\_Resume, the next file will

#### E) STOPPING THE PREVIEW AND RECORDING

```
VideoGrabber.Stop();
```

#### F) SETTING A NEW FILE NAME BEFORE RESTARTING THE VIDEO

bool OpenInPausedState = false; // to record to the new file immediately, otherwise the current file is closed and the recording to the next file will start when invoking Encoder\_Resume

```
VideoGrabber.Encoder_NewOutputFile(m_EncoderInstanceID,
```

```
"c:/folder/anothervideoclip.mp4", OpenInPausedState);
VideoGrabber.StartPreview();
```

Created with the Standard Edition of HelpNDoc: [Eliminate the Struggles of Documentation with a Help Authoring Tool](#)

## Pause/resume during recording

### Pause/resume during recording

[Prev](#)
[Next](#)

Pausing and resuming the recording

#### Description

#### Pause/resume during recording

It is possible to pause / resume the recording if [RecordingCanPause](#) has been enabled before invoking [StartRecording](#).

The recording can be paused by invoking [PauseRecording](#) , and resumed by invoking [ResumeRecording](#).

The [OnRecordingPaused](#) event notifies that the pause recording is completed (when the recording is resumed, a [OnRecordingStarted](#) event notification occurs).

Remark:

*While the recording is paused the video source goes on streaming and previewing. If you need to pause the video source as well look at the "pausing/resuming the whole recording graph" chapter below.*

#### Appending the video when resuming the recording or generating a new clip

- if [RecordingPauseCreatesNewFile](#) is **disabled**, the recording is appended to the current video clip.

#### Splitting the resume/pause sequences in several files

- if [RecordingPauseCreatesNewFile](#) is **enabled**, a new clip is generated automatically for each resume/pause sequence.

(if the [RecordingFileName](#) property is empty a new file name is generated automatically depending on the [AutoFileName](#) setting, otherwise the recording file name specified just before invoking [RecordingPauseCreatesNewFile](#) is used)

The new file is not created when resuming the recording, but before when the recording has previously been paused by invoking [PauseRecording](#).

Therefore, the file is already created and opened when invoking [ResumeRecording](#), "ready to go", so by this way frames will not be dropped by loosing time to create the new file.

So if the recording is stopped before invoking [ResumeRecording](#), the new empty file (that has just been created before, when [PauseRecording](#) has been invoked) is deleted.

*Note: the pause / resume while compressing on the fly may not work properly with some video compressors.*

#### Pausing/resuming the whole recording graph

During recording, when invoking [PauseRecording](#) / [ResumeRecording](#), the graph goes on running and previewing (only the recording is paused)

If you need to pause /resume the whole graph during the recording, invoke [PausePreview](#) / [ResumePreview](#) instead.

This can be useful e.g. to apply overlays on a video clip by recording it to a new file with the following settings:

```
VideoGrabber.VideoSource = vs_VideoFileOrURL
VideoGrabber.VideoSource_FileOrURL = name of the source clip
VideoGrabber.StartRecording ()
...
```

- then invoke later VideoGrabber.[PausePreview](#) to pause the recording of the video clip, so you can prepare and/or update the overlays,
- then invoke later VideoGrabber.[ResumePreview](#) to go on recording from the location where the clip has been paused.

#### See Also

[Datastead Encoder Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveAllInstances](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## Preroll recording (Backtimed recording)

### Preroll recording (Backtimed recording)

[Prev](#)

[Next](#)

Pre-roll recording (starting with a back-timed delay)

#### Description

#### Preroll recording

It is possible to record the file with a back-timed amount of video frames (pre-roll). This property is activated by specifying a [RecordingBacktimedFramesCount](#) value (greater than 0).

This feature is useful when the recording is started by an event (e.g. motion detected) and you want to start the recording of the AVI file a few moments before this event occurs.

When [RecordingBacktimedFramesCount](#) has a value > 0, TVideoGrabber uses a FIFO (first in, first out) memory buffer to store the latest video frames.

This property can be used e.g. with the [motion detection](#) features. When a motion is detected and the recording starts, this property allows you to have at the beginning of the recording the few moments before the motion detection event occurs.

By this way recording is shifted back in time and starts at the "current - n" video frame.

When starting the recording, the normal video frames are streamed "normally" until the buffer is filled out. Then, when the buffer is filled out, the [OnBacktimedFrameCountReached](#) event occurs to let you know that now current video frames are replaced by back-timed video frames..

Note that when invoking StopRecording the recording stops "as is", this means that the latest video frames in the buffer will be missing (only the backtimed frames are recorded), unless you stop the recording at the desired time + the delay corresponding to the frames in the buffer.

**Note:** be careful when assigning a number of frames to this property, this means that a corresponding buffer of video frames will be created in memory during recording.  
E.g. 100 frames in 320 x 240 x 24 bits = 320 x 240 x 3 x 100 = 22,500 Kb = 22 Mb of memory required during recording.

#### Requirements

- the [Frame grabber](#) must be set to fg\_BothStreams
- the [RecordingInNativeFormat](#) must be disabled.



- the [RecordingMethod](#) must be set to rm\_AVI.

### Implementation

- to preview the back-timed video stream, set [FrameGrabber](#) = fg\_BothStream. If you prefer to preview the normal video stream, set [FrameGrabber](#) = fg\_CaptureStream.
- set [RecordingInNativeFormat](#) = false,
- set [HoldRecording](#) to true to start recording in preview mode and hold AVI writing,
- call [StartRecording](#) to put the recording in preview mode and start to fill the buffer,
- when the [OnBacktimedFrameCountReached](#) event occurs (1), the buffer is filled out
- then invoke [ResumeRecording](#) to really start the AVI recording.

(1) when this event occurs, back-timed video frames start replacing normal video frames in the captured stream.

### quick test code:

```
procedure TForm1.BtnPrepareRecordingClick(Sender: TObject);
begin
    VideoGrabber.OnRecordingReadyToStart := nil;
    VideoGrabber.FrameGrabber := fg_BothStreams;
    VideoGrabber.FrameRate := 10; // 10 fps
    VideoGrabber.RecordingBacktimedFramesCount := 30; // 30 frames = e.g. 3 seconds at 10
    VideoGrabber.RecordingInNativeFormat := False;
    VideoGrabber.HoldRecording := True;
    VideoGrabber.StartRecording;
end;

procedure TForm1.BtnStartRecordingClick(Sender: TObject); // wait for the buffer to fill
begin
    VideoGrabber.ResumeRecording;
end;

procedure TForm1.BtnStopRecordingClick(Sender: TObject);
begin
    VideoGrabber.StopRecording;
end;
```

---

Created with the Standard Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

---

## Playing a video clip during recording

### Playing a video clip during recording

[Prev](#)
[Next](#)

How to play a video clip while it is recording?

#### Description

It is possible to play a video clip in a 2nd instance of TVideoGrabber while it is recording in ASF format a 1st instance as follows:

#### In the 1st instance of TVideoGrabber:

1.
  - set CompressionMode = cm\_NoCompression
  - select the "ASF" recording method (RecordingMethod = rm\_ASF),

2. invoke StartRecording.

The video clip starts recording, by using the name specified in the RecordingFileName property (unless

you left it blank, in this case a recording file name is generated automatically and returned by the OnRecordingStarted event).

#### In the 2nd instance of TVideoGrabber:

1. set VideoSource\_FileOrUrl = the path / name of the file currently recording the 1st instance
2. set VideoSource = vs\_VideoFileOrURL
3. invoke StartPreview

and the video clip currently recording will start playing in the 2nd instance of TVideoGrabber.

#### This can be tested quickly by running MainDemo.exe as follows:

a)

- run a first instance of MainDemo.exe,
- click on the "Recording" tab,
- select "ASF" as recording method (on the right of the tab),
- click on "start recording"

The name of the file currently recording will appear in the memo on the left, copy it to the clipboard.

b)

- run a 2nd instance of MainDemo.exe,
- in the "video source = file or URL" group, click on the "open" button and paste the name of the file currently recording
- browse the "VIDEO SOURCE" list, and select "video file or URL"
- then click on the "Start" preview button

and the video clip currently recording in the 1st instance of MainDemo.exe will start playing in the 2nd instance.

*Remark: the clip played while recording is not seekable, because the total duration will be determined only when the recording completes.*

---

Created with the Standard Edition of HelpNDoc: [Free PDF documentation generator](#)

---

## Dropped frames and audio/video sync problems

### Dropped frames and audio/video sync problems

[Prev](#)
[Next](#)

Dropped frames and audio/video sync problems

#### Description

#### DROPPED FRAMES

The live video and audio sources stream data at a fixed frame rate, and this means that the processor must have finished processing the previous frame when the next frame arrives, otherwise the frame is dropped and replaced by a blank frame.

The dropped frame count can be retrieved by reading the [DroppedFrames](#) property.

Because reading the dropped frames needs more or less CPU (depending of the video capture device), the number of dropped frames is not updated for every frame, but refreshed periodically according to the [DroppedFramesPollingInterval](#) property (specified in seconds).

E.g:

[DroppedFramesPollingInterval](#) = 20 polls the dropped frames count every 20 seconds.

[DroppedFramesPollingInterval](#) = 0 disables the polling of dropped frames (recommended to save CPU).

#### Disk access and preallocated files

During recording, one major problem is the block allocation when the operating system extends the recording file. When allocating new blocks, the operating system places significant demands on the CPU/disk resources, and this can cause dropped frames.



This problem occurs mainly when capturing uncompressed video in AVI format. It is less significant when using a video compressor on the fly (or when recording in ASF format), because the file size is 10 times or more smaller.

The best solution to avoid this problem is to record to a SSD (if a lot of disk space is needed for multiple recording, once recorded move the video clip from the SSD to a normal HDD, so the SSD is used only as temporary disk just for the recording).

Another solution is to use an huge preallocated capture file, larger than the amount of data to capture. See the "**preallocated capture file**" section in the [AVI recording](#) chapter.

FAT 32 is not efficient for large capture files, generating large amounts of dropped frames when new disk blocs are allocated.

On Windows 2000 and Windows XP platforms, always choose NTFS partitions on disks with fast access time and cache memory.

### Frame grabber

The frame grabber adds an additional step in the stream processing, and therefore requires more CPU. If you don't need to capture frames or perform text or graphics overlays during recording, simply disable it by setting [FrameGrabber](#) = fg\_Disabled before invoking [StartRecording](#).

### Audio rendering

During recording of video + audio streams, disable if possible [AudioDeviceRendering](#), that renders sound in the loudspeakers. This will let TVideoGrabber capture the video and audio streams without having to split the audio stream. Audio rendering must be disabled before invoking [StartRecording](#).

### Video Display

In order to save CPU you can disable the video window during the recording. Simply set VideoRenderer = vr\_None before invoking [StartRecording](#).

### Video size and frame rate

Of course, larger is the video size and higher is the frame rate, higher is the CPU and disk consumption. Reducing the video size to 320x240 and the frame rate will help to fixing audio sync problems.

---

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

---

## AES Encryption

---

Created with the Standard Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

---

### AES Encryption Overview

#### AES Encryption Overview

[Prev](#)
[Next](#)

#### AES Encryption Overview

##### Description

##### It is possible to :

- encrypt a recording in real-time through AES with a 128-bit or 256-bit key to any video container (AVI, MP4, MKV, etc...)
- play and seek the encrypted video with [OpenPlayer\(\)](#). The decryption is performed in real-time.
- batch-encrypt a non-encrypted video clip to produce an encrypted video clip
- batch-decrypt an encrypted video clip to produce an unencrypted video clip

**The encryption/decryption require the optionals** Datastead RTSP/RTMP/HTTP/ONVIF Source 9.2 or above and/or Datastead Multipurpose Encoder 3.2 or above.

- **the encryption is activated** by passing an encryption key to [SetEncryptionKey](#).

- the decryption is activated by passing the the key used for encryption to [SetDecryptionKey](#).

Both features can be deactivated by passing an **empty key**.

**The Datastead RTSP/RTMP/HTTP/ONVIF Source can:**

- encrypt in real-time the recording IP cameras or other URL sources in native format (without decompression/recompression) by setting **SetEncryptionKey**.
- play and seek in real-time any video previously encrypted through the player features. The decryption is enabled with **SetDecryptionKey**.
- batch encrypt existing video files with `Encrypt_File`
- batch decrypt encrypted video files with `Decrypt_File`

**The Datastead Multipurpose Encoder can:**

- encrypt in real-time during recording the audio/video source being recorded, excepted the "recording of IP cameras and other URL sources in native format" case above that is done directly by the Datastead RTSP/RTMP/HTTP/ONVIF Source (described above)
- batch encrypt existing video files with `Encrypt_File`
- batch decrypt encrypted video files with `Decrypt_File`

---

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

## Key

### Key

[Prev](#)

[Next](#)

Encryption/decryption Key

#### Description

The encryption/decryption key is set by passing a string of 16 characters length (128-bit key) or 32 characters length (256 bits).

For security reasons this key is not stored directly as a string in memory by `SetEncryptionKey` or `SetDecryptionKey`.

Note: **do not loose the key**. Once the file encrypted, without the key it is impossible to recover the file.

---

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

---

## Realtime Encryption or Decryption

### Realtime Encryption or Decryption

[Prev](#)

[Next](#)

Realtime Encryption or Decryption

#### Description

##### Enabling the realtime encryption

- to enable the **encryption** during recording, invoke [SetEncryptionKey](#) and pass an encryption key of your choice (**that MUST be 16 characters or 32 characters length**), then start the recording as usual
- to enable the **decryption** of an encrypted video during playback, invoke [SetDecryptionKey](#) and pass **the key that has been used for the encryption**, then open the clip for playback as usual

Note: the ASF recording does not yet support encryption for the moment.

#### See Also

[TEncryptionMethod](#) [Decrypt\\_File](#) [Encrypt\\_File](#) [EncryptionMethod](#) [SetDecryptionKey](#) [SetEncryptionKey](#)

---

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

## Batch Encryption or decryption

### Batch Encryption or decryption

[Prev](#)

[Next](#)

Batch Encryption or decryption

#### Description

- to batch-encrypt a video file, invoke [SetEncryptionKey](#), then invoke [Encrypt\\_File](#)
- to batch-decrypt a video file, invoke [SetDecryptionKey](#), then invoke [Decrypt\\_File](#)

## Disabling the encryption/decryption

### Disabling the encryption/decryption

[Prev](#)

[Next](#)

Disabling the encryption/decryption

#### Description

- after finishing an encryption job (e.g. recording or encrypting an existing video clip), **to disable the encryption invoke [SetEncryptionKey](#)** and pass an empty string.
- after finishing an decryption job (e.g. playback or decryption of an encrypted video clip), **to disable the decryption invoke [SetDecryptionKey](#)** and pass an empty string.

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## Frame capture

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

### Frame capture features

#### Frame capture features

[Prev](#)

[Next](#)

Frame capture overview.

#### Description

##### Frame capture

It is possible to capture video frames during preview, recording and play back when the [frame grabber](#) is enabled.

The frame grabber can be inserted on the preview stream, capture stream, or both. See the [FrameGrabber](#) property.

Frames may be captured to:

- memory bitmaps,
- BMP files,
- JPEG files,
- clipboard (CF\_BITMAP)

one by one or automatically in burst mode.

#### I. DIRECT COPY OF THE BITMAP HANDLE (OR DIRECTLY THE BITMAP BITS ARRAY) OF THE CURRENT VIDEO FRAME FROM THE [OnFrameBitmap](#) EVENT

For each video frame travelling through the graph, the [OnFrameBitmap](#) event occurs and allows you to capture directly the bitmap bits.

This is the method that requires the less CPU, however as the [OnFrameBitmap](#) event is called directly from a thread (1), the bitmap handle and the bitmap bits are valid only when the event occurs so you have to copy them to another buffer immediately.

Noet that this method can slow down the graph and/or produce a jerky preview, because the processing time added in the event is included in the frame processing time, so if the time to process 2 frames becomes longer than the frame rate frames will be dropped.

So it recommended rather when the graph is previewing (by invoking [StartPreview](#)), in this case dropped frames are less critical than during a recording (this could produce a jerky recording) or during playback (that may appear jerky).

Sample code can be found in the *OnFrameBitmap* event of the *MainDemo* project (seach "FrameBitmap" in the *MainForm*)

(1) the event can be synchronized with the Windows main thread if the [OnFrameBitmapEventSynchronone](#) property is enabled.

## II. DIRECT FRAME CAPTURE OF THE CURRENT (OR PREVIOUS FRAMES)

You can invoke [GetLastFrameAsHBitmap](#) (all versions) or [GetLastFrameAsTBitmap](#) (only with the Delphi/C++Builder versions), to get a copy of the current (or previous) frame with the size of your choice, and with or without overlays.

This way does not slow down the graph and does not generate dropped frames.

A sample code is shown in the *MotionDetected* event of the *MainDemo* project, that displays a copy of the last frame detected with motion in the "Overlay" tab.

## III. FRAME CAPTURE OF THE CURRENT FRAME (OR NEXT FRAMES AUTOMATICALLY IN BURST MODE)

by using [CaptureFrameTo](#) (or enabling the burst mode) + the [OnFrameCaptureCompleted](#) event

This way does not slow down the graph and does not generate dropped frames.

### With or without overlays

If the application performs graphic or text overlays, by default the frames are captured with overlays. To capture the frames without overlays, enable the [FrameCaptureWithoutOverlay](#) property.

### ASYNCHRONOUS VS. SYNCHRONOUS FRAME CAPTURE

#### A) Asynchronous frame capture

##### to capture frames one by one

Invoke [CaptureFrameTo](#) . The function send a "capture order" and exits immediately. Then the captured frame is returned by the [OnFrameCaptureCompleted](#) event.

##### to capture frames automatically in burst mode

The burst mode allows to capture frames automatically. It is activated by enabling the [BurstMode](#) property.

Frames can be captured to TBitmap, BMP file or JPEG file according to the [BurstType](#) property.

Frames are captured indefinitely if [BurstCount](#) = 0, otherwise frame capture stops when a [BurstCount](#) number of frames have been captured.

If [BurstInterval](#) > 0, a [BurstInterval](#) number of frames will be skipped between each frame captured.

The burst mode is activated by enabling [BurstMode](#) .

The burst mode is deactivated:

- by disabling [BurstMode](#) ,
- if [BurstCount](#) > 0, when a [BurstCount](#) number of frames have been captured.

When capturing video frames to BMP or JPEG files:

- if no file name is specified when invoking [CaptureFrameTo](#) or [CaptureFrameSyncTo](#), the file name is generated automatically (1).
- in burst mode file names are always generated automatically (1).

(1) according to [StoragePath](#) , [AutoFileName](#) and [AutoFilePrefix](#) .

### Frame capture of unmodified frames when performing frame overlay

When drawing over video frames by using the [Frame Overlay](#) feature, by default the frame capture occurs AFTER frames have been overlayed (objects drawn over video frames appear on captured frames). Enable the [FrameCaptureWithoutOverlay](#) property to capture unmodified frames before they are overlayed.

#### Example of asynchronous frame capture

The method is the same as described above, excepted that you have to set [UseClock](#) = false before opening the clip, to play it at the maximal speed.

E.g.:

```
VideoGrabber.BurstMode = true
VideoGrabber.BurstCount = 5
VideoGrabber.BurstInterval = 10
VideoGrabber.BurstType = fc_BmpFile
VideoGrabber.PlayerFileName = "... the file name of my video clip..."
VideoGrabber.UseClock = false
VideoGrabber.OpenPlayer
```

### Size of the captured frames

#### Default size

- by default, the size of the captured frames is the current size of the video frames (returned by the [VideoWidth](#) and [VideoHeight](#) properties).

#### Zoomed size

- it is possible to zoom the captured frames by specifying a [FrameCaptureZoomSize](#) percentage. The default value is 100 (100%)

#### Custom size

- it is possible to stretch the captured frames to specified width, height or both by assigning a value to [FrameCaptureWidth](#) and/or [FrameCaptureHeight](#). The default value for both is -1 (disabled)

NB: if a value  $\neq$  100 is assigned to [FrameCaptureZoomSize](#), [FrameCaptureWidth](#) and [FrameCaptureHeight](#) are ignored.

### B) Synchronous frame capture through the frame grabber

The function will wait for the frame capture to be completed before returning.

invoke [CaptureFrameSyncTo](#) . The function waits for the frame to be captured. The frame is returned the frame through the [OnFrameCaptureCompleted](#) , and then function returns true upon success, exactly like for the asynchronous [CaptureFrameTo](#) described above.

### C) Synchronous frame capture with the frame grabber disabled

It is possible to capture the frame being displayed in the video renderer by invoking [CaptureFrameRenderedTo](#), allowing to disable the frame grabber, to save CPU load and memory.

### HOW TO CAPTURE A SUB-RECTANGLE OF THE VIDEO FRAME

To capture a sub-rectangle, invoke [SetFrameCaptureBounds](#) (Left, Top, Right, Bottom) before invoking [CaptureFrameTo](#) or [CaptureFrameSyncTo](#). (to retrieve the current size of the video frame use [VideoWidth](#) and [VideoHeight](#))

### HOW TO STRETCH THE CAPTURED VIDEO FRAME

To resize the video frame as desired, specify a [FrameCaptureWidth](#) and/or [FrameCaptureHeight](#) value

### HOW TO ZOOM THE CAPTURED VIDEO FRAME

To resize the captured frame, specify to [FrameCaptureZoomSize](#) a zooming percentage.

### TOP-DOWN AND LEFT-RIGHT FRAME CAPTURE

This can be easily achieved by invoking [SetFrameCaptureBounds](#) and specifying the bottom instead of the top, and/or the left instead of the right.  
(to retrieve the current size of the video frame use [VideoWidth](#) and [VideoHeight](#))

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## Graphic and text overlays

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

### Setting and retrieving the overlay properties

#### Setting and retrieving the overlay properties

[Prev](#)
[Next](#)

Graphic and text overlays

#### Description

TVideoGrabber includes build-in functions to manipulate text and image overlays.

Several text overlays or image overlays can be applied concurrently on the same video stream, **each overlay is identified by its index, in the 0..n-1 range**.

The indexes of the overlays are independent, it is possible to use a text overlay with index 0 and an image overlay with index 0 at the same time.

#### The overlays can be set in 2 ways:

- either by invoking `SetOverlay_propertyname` (index, value) or `GetOverlay_propertyname` (Index)
- either by setting the overlay index selector first, then setting or reading the property name

#### 1. THE -STRAIGHT- FUNCTION WAY (recommended) e.g.:

```
VideoGrabber.SetTextOverlay_String (0, "first overlay");
VideoGrabber.SetTextOverlay_Left (0, 10);
VideoGrabber.SetTextOverly_Enabled (0, true);
```

```
VideoGrabber.SetTextOverlay_String (1, "2nd overlay");
VideoGrabber.SetTextOverlay_Left (1, 200);
VideoGrabber.SetTextOverly_Enabled (1, true);
```

#### Text overlays

- the text overlays can be set by invoking `SetTextOverlay...` (text overlay index, value) functions, e.g. [SetTextOverlay\\_Enabled](#) (0, true) for the 1st text overlay
- the text overlay properties can be read by invoking `GetTextOverlay...` (text overlay index) functions, e.g. [GetTextOverlay\\_Left](#) (3) for the 4th text overlay

#### Image overlays

- the image overlay settings can be set by invoking `SetImageOverlay...` (image overlay index, value) function, e.g. [SetImageOverlay\\_Transparent](#) (2, true) for the 3rd image overlay
- the image overlay settings can be read by invoking `GetImageOverlay...` (image overlay index) functions, e.g. [GetImageOverlay\\_TopLocation](#) (1) for the 2nd image overlay)

**2. THE "PROPERTIES" WAY (not recommended when using many overlays) e.g.:****VideoGrabber.TextOverlay\_Selector = 0**

VideoGrabber.TextOverlay\_String = "first overlay"

VideoGrabber.TextOverlay\_Left = 10

VideoGrabber.TextOverlay\_Enabled = true

**VideoGrabber.TextOverlay\_Selector = 1**

VideoGrabber.TextOverlay\_String = "2nd overlay"

VideoGrabber.TextOverlay\_Left = 200

VideoGrabber.TextOverlay\_Enabled = true

**Text overlays**

The text overlays can be set and retrieved:

- by first setting the [TextOverlay\\_Selector](#) property
- then by setting or reading the TextOverlay\_... properties, e.g. [TextOverlay\\_Left](#)

**Image overlays**

The image overlays can be set and retrieved:

- by first setting the [ImageOverlaySelector](#) property
- then by setting or reading the ImageOverlay\_... properties, e.g. [ImageOverlay\\_LeftLocation](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation with HelpNDoc's Project Analyzer Features](#)

**Frame overlay vs window overlay****Frame overlay vs window overlay**[Prev](#)[Next](#)**Frame overlay vs Window Overlay****Description****FRAME OVERLAY vs WINDOW OVERLAY**

It is possible to apply the overlays:

**A) over the video frames**(the [frame grabber](#) must be enabled).

- if the video frames are stretched, the overlays are stretched too
- if the video source is being recorded and [RecordingInNativeFormat](#) is disabled, the overlays are applied to the recording.

To apply the overlay over the video frame, set the "target display" property to **-1** (*default value anyway*) by invoking [SetTextOverlay\\_TargetDisplay](#) (overlay index, **-1**), or [SetImageOverlay\\_TargetDisplay](#) (overlay index, **-1**)

**B 1) over the display video window(s)**

- the size of the overlays is fixed and not stretched when the video frame or video window is resized
- if 2 or more video windows are used for the same video stream (enabled by the DualDisplay... properties), it is possible to apply the overlay to only one of the video windows.

To apply the overlay over ALL the video windows, set the "target display" property to **-2** by invoking [SetTextOverlay\\_TargetDisplay](#) (overlay index, **-2**), or [SetImageOverlay\\_TargetDisplay](#) (overlay index, **-2**)

To apply the overlay over only 1 of the video windows, set the "target display" property to the index of the video window in the 0..n-1 range, by invoking e.g. [SetTextOverlay\\_TargetDisplay](#) (overlay index, **0**), or [SetImageOverlay\\_TargetDisplay](#) (overlay index, **1**)

The video windows indexes are assigned as follows:

- the **default video window** ("Display\_..." settings) corresponds to the **index 0**
- the **2nd video window** ("DualDisplay\_..." settings) corresponds to the **index 1**



- the 3rd video window corresponds to the index 2  
and so on...

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## Overlays and aspect ratio

### Overlays and aspect ratio

[Prev](#)

[Next](#)

About overlays and aspect ratio

#### Description

#### About overlays and aspect ratio

Some video sources or video clips output a video size that does not match the display size.

E.g. the size of the video frames is 720x480 and the display aspect ratio is 16:9, so the video frames are stretched to 720x405 in the video window.

By default, the [AdjustOverlayAspectRatio](#) property is enabled and prevents the overlays to be distorted when the video frames are stretched into the video window to be displayed at their correct display aspect ratio.

Disabling this property may save CPU, if no matter the display aspect ratio of the overlays.

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

## Image overlays

### Image overlays

[Prev](#)

[Next](#)

#### Image overlays

#### Description

#### Image overlays over video frames

#### Overview

It is possible to overlay one or several images at the same time over the video frames (using more than one image overlay is enabled by the ImageOverlaySelector property described in this chapter).

The origin of the image can be:

- an image file
- a bitmap handle
- a TImage component (not available in OCX versions)
- a TBitmap component (not available in OCX versions)

For image files, the following format are supported:

BMP  
GIF  
JPEG  
EXIF  
PNG  
TIFF  
WMF  
EMF  
EMF+  
EMF+ Dual

Unlike the [logo](#), that is displayed in the video window when the component is **not running**, the image will be overlaid over the video frames when the component is previewing, recording or playing back.



## Source of the image overlay

The image can be loaded from various image source formats by invoking:

A) straight method (overlay index passed as parameter):

[SetImageOverlayFromImageFile2](#) (Index, filename): loads the overlay #index with the specified file image

[SetImageOverlayFromHBitmap2](#) (Index, bitmapHandle): loads the overlay #index with the specified bitmap handle

[SetImageOverlayFromTBitmap2](#): loads the overlay #index with the specified TBitmap Object (Delphi and C++Builder)

[SetImageOverlayFromTImage2](#): loads the overlay #index with the specified TImage Object (Delphi and C++Builder)

B) image overlay selector method: (see [Setting and retrieving the overlay properties](#))

- first set [ImageOverlaySelector](#) with the desired index to select the overlay

- then invoke one of the following functions:

[SetImageOverlayFromImageFile](#) (filename): loads the current overlay with the specified file image

[SetImageOverlayFromHBitmap](#) (bitmap handle): loads the current overlay with the specified bitmap handle

[SetImageOverlayFromTBitmap](#) (TBitmap object): loads the current overlay with the specified TBitmap Object (Delphi and C++Builder)

[SetImageOverlayFromTImage](#) (TImage object): loads the current overlay with the specified TImage Object (Delphi and C++Builder)

## Activation/deactivation of the image overlay

The image overlay may be enabled or disabled on the fly by enabling or disabling

[SetImageOverlay\\_Enabled](#) (index, value).

## Location of the image overlay

The x,y location of the overlay over the video frames is specified by [SetImageOverlay\\_LeftLocation](#) and [SetImageOverlay\\_LeftLocation](#).

ImageOverlay\_LeftLocation must be in the [0..video width] range.

ImageOverlay\_TopLocation must be in the [0..video height] range.

## Size of the image overlay

The size to which the image will be stretched can be specified (in pixels) by [SetImageOverlay\\_Width](#) and [SetImageOverlay\\_Height](#).

To use the real width of the image overlay set ImageOverlay\_Width = -1

To use the real height of the image overlay set ImageOverlay\_Height = -1

## Stretching the image overlay to the video size

When [SetImageOverlay\\_StretchToVideoSize](#) is enabled, this property stretches the image overlay to the video size (in this case the image overlay location, width and height settings above are ignored)

## Alpha-blending (semi-transparency)

The alphablending is enabled by [SetImageOverlay\\_AlphaBlend](#).

The value of the alpha blending (in the [0..255] range) must be specified by

[SetImageOverlay\\_AlphaBlendValue](#).

## Color-based transparency

The color-based transparency is enabled by [SetImageOverlay\\_Transparent](#).

If [SetImageOverlay\\_UseTransparentColor](#) is disabled, the default color (background color) of the image is used.

If [SetImageOverlay\\_UseTransparentColor](#) is enabled, the color specified by [SetImageOverlay\\_TransparentColorValue](#) will be used.

### Specifying all the properties at the time

You can invoke [SetImageOverlayAttributes](#) (instead of setting the image overlay properties one by one) to set all the settings above from a single function call.

### Applying several image overlays concurrently

The [ImageOverlaySelector](#) property lets you use up to 1000 image overlays concurrently. This is just a theoretical limit, the corresponding resources will be allocated only when you activate the corresponding selector.

You have just to specify the image overlay number before invoking [SetImageOverlay...](#) and [ImageOverlayEnabled](#).

E.g.:

#### **- loading and showing myimage1.gif:**

```
ImageOverlaySelector = 0
SetImageOverlayFromAnyFile ("myimage1.gif")
ImageOverlayEnabled = true
```

#### **- loading and showing myimage2.png at the (10, 20) location with a default size:**

```
ImageOverlaySelector = 1
SetImageOverlayFromAnyFile ("myimage2.png")
ImageOverlay_LeftLocation = 10
ImageOverlay_TopLocation = 20
ImageOverlay_Width = -1
ImageOverlay_Height = -1
ImageOverlayEnabled = true
```

#### **- hiding myimage1.gif**

```
ImageOverlaySelector = 0
ImageOverlayEnabled = false
```

#### **- modifying the location of myimage2.png at (40, 60) and resizing it at 120x90:**

```
ImageOverlaySelector = 1
ImageOverlay_LeftLocation = 40
ImageOverlay_TopLocation = 50
ImageOverlay_Width = 120
ImageOverlay_Height = 90
```

#### **- showing myimage1.gif semi-transparent**

```
ImageOverlaySelector = 0
ImageOverlayEnabled = true
ImageOverlay_AlphaBlend = true
ImageOverlay_AlphaBlendValue = 140
```

### See Also

[GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)

## Text overlays

### Text overlays

[Prev](#)
[Next](#)

Text overlays

#### Description

#### Text over video frames

- set the [SetTextOverlay\\_String](#) property with the text string to write ([string options](#)),
- set the [SetTextOverlay\\_Font](#) color and font size,
- set the [SetTextOverlay\\_Transparent](#) text transparency.
- set the [SetTextOverlay\\_BkColor](#) background color (useful only when TextOverlay\_Transparency is disabled)
- set the [SetTextOverlay\\_Align](#) text alignment,
- set the [SetTextOverlay\\_Left](#) , [SetTextOverlay\\_Right](#) and [SetTextOverlay\\_Top](#) text location,
- invoke [SetTextOverlay\\_Enabled](#) to activate the text overlay.

Remark: a shadow can be draw under the text overlay when [SetTextOverlay\\_Shadow](#) is enabled, according to the [SetTextOverlay\\_ShadowDirection](#) and [SetTextOverlay\\_ShadowColor](#) values.

#### Alpha blending

To enable the alpha blending of the text overlays:

- enable [SetTextOverlay\\_AlphaBlend](#)
- set [SetTextOverlay\\_AlphaBlendValue](#) with a transparency value in the 0..255 range

#### Text scrolling

The text scrolling can be activated by enabling the [SetTextOverlay\\_Scrolling](#) property.  
The scrolling speed is adjusted with the TextOverlay\_Scrolling\_Speed:

TextOverlay\_Scrolling\_Speed > 0 -> scrolling from right to left

TextOverlay\_Scrolling\_Speed < 0 -> scrolling from left to right

TextOverlay\_Scrolling\_Speed = 0 -> stops the scrolling

When a scrolling is completed (when the last character disappear from the video frame) the [OnTextOverlayScrollingCompleted](#) event occurs.

The TextOverlayIndex parameter of the OnTextOverlayScrollingCompleted event returns the index of the related text overlay.

#### Custom fonts

You can create a fully customised font by invoking [TextOverlay\\_CreateCustomFont](#) or [TextOverlay\\_CreateCustomFont2](#)

E.g.:

```
VideoGrabber.TextOverlay_CreateCustomFont (16, 30, 0, 0, 200, true, false, false, 2, 0, 0, 0, 0, "Wingdings")
```

#### Using several text overlays, each one having its own set of font / color / background

You can define up to 1000 different groups of text overlays settings, that will be drawn at the same time over video frames, each one having different TextOverlay\_... values.

Each group is selected by setting [TextOverlay\\_Selector](#) to the corresponding value (in the [0..999] range).  
By default the TextOverlay\_... settings concern the 1st group (TextOverlay\_Selector = 0).

At startup TVideoGrabber does not allocate the resources for 1000 sets of text overlays. By default only one

font and one parameter sets are allocated (corresponding to `TextOverlay_Selector = 0`). When using more than one text overlay, the required resource allocation is done when the `TextOverlay_Selector` property receives a greater value than the latest highest value (see *the remark below*).

To use more than one group simply set [TextOverlay\\_Selector](#) with the index of the group (in the 0..999 range) before reading or setting the corresponding `TextOverlay_...` properties.

Look at the [TextOverlay\\_Selector](#) property for same code.

**Remark:**

*when using more text overlays, the additional memory and fonts required are allocated when a new `TextOverlay_Selector` value set (and greater than the highest value previously used).*

*E.g. if the previous value of `TextOverlay_Selector` was 5 and if you set now `TextOverlay_Selector = 20`, this will allocate 15 new fonts and parameter sets, one for each new text overlay (for a total of 20 fonts and parameter sets allocated).*

*So if you need e.g. only 5 different text overlays in your app (in this case `TextOverlay_Selector` will be used in the [0..4] range), be sure to NEVER set `TextOverlay_Selector` to a value GREATER THAN 4, to prevent allocating more resources than needed (by allocating useless text overlay fonts and parameter sets).*

---

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

---

## Graphic overlays

### Graphic overlays

[Prev](#)
[Next](#)

#### [Example](#)

Graphics and text overlays.

#### Description

##### Frame overlay

When the [frame grabber](#) is enabled, it is possible to perform graphics or text overlay over uncompressed RGB video frames during preview, recording or playback.

For the graphics and text overlays to **be applied over the video frames recored in the AVI** during recording, the [RecordingInNativeFormat](#) property must be **disabled**.

#### Getting direct access to the current video frame bitmap

You can get direct access to each video frame by using the [OnFrameBitmap](#) event, that occurs for each video frame. Click [here](#) for more information about this event.

*You will find the sample code in the "MainForm" of the MainDemo project included in te package -> search for "FrameBitmap"*

**From this event you should not perform any actions with the potential to block:**

1. do not hold a critical section or wait on another thread,
2. do not call any GDI or USER32.DLL APIs that might cause a window to move,
3. do not invoke component properties that send messages, e.g. do not read the `ItemIndex` property of a `ListBox` component.

If you need to read a such property, set an intermediary variable when the property changes, and then read the intermediary variable from the `OnFrameOverlay...` event.

#### Restrictions

- a. when the video is recorded in "native" format ([RecordingInNativeFormat](#) = true), drawing over video frames will be **visible** on the preview stream but **not on the AVI stream** (native means "unmodified"),
- b. during preview or recording of video capture boards having **a video port** (e.g. like ATI Radeon), drawing over video frames will **not be visible** on the **preview stream** when the [video port](#) is **enabled**.

Therefore:

1. for the overlays to be visible on the preview window, disable the [video port](#) (but depending of the video capture device the preview could be jerky when previewing large video sizes (or in full screen mode),
2. in order to include the frame overlay in the recorded stream, disable [RecordingInNativeFormat](#) (however

this requires higher CPU consumption during recording, because the video stream can have to be uncompressed before applying the overlays).

### Drawing bitmaps over video frames

Bitmaps can be drawn "as is" or stretched over video frames by invoking the [DrawBitmapOverFrame](#) function.

This function includes the transparency and alpha blending features.

It MUST be called only from the [OnFrameBitmap](#) event, that occurs for each video frame.

### Drawing a shape over the video frames *(not available in the OCX versions)*

To draw a shape over video frames:

- put a TShape component on the form,
- assign the TShape component to the ShapeOverlay property,
- disable the TShape's visible property to avoid the shape being visible at runtime,
- enable [ShapeOverlayEnabled](#) to activate drawing shapes over video frames,
- at runtime, define the TShape's Left, Top, Width and Height properties, that will be used by TVideoGrabber to position and resize the shape over video frames.

### Drawing several shapes over video frames *(not available in the OCX versions)*

To draw several shapes over video frames:

- put TShape components on the form (e.g. Shape1 and Shape2),
- call VideoGrabber1.[ShapeOverlayList](#) (Shape1, true) to add the 1st shape to the TVideoGrabber's shape list,
- call VideoGrabber1.[ShapeOverlayList](#) (Shape2, true) to add the 2nd shape to the TVideoGrabber's shape list,
- disable the Shape1 and Shape2 visible properties to avoid the shape being visible at runtime,
- enable [ShapeOverlayEnabled](#) to activate drawing shapes over video frames,
- at runtime, define the Shape1 and Shape2's Left, Top, Width and Height properties, that will be used by TVideoGrabber to position and resize the shapes over video frames.

Note: Shapes are not drawn if their *Enabled* property is disabled.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

## Chroma key

### Chroma key

[Prev](#)

[Next](#)

Chroma Key

#### Description

#### Chroma Key

The chroma key feature is activated as follows:

- load a background image with [SetImageOverlayFromImageFile](#) (or another SetImageOverlayFrom... function)
- enable the [ImageOverlayEnabled](#) property
- enable the [ImageOverlay\\_ChromaKey](#) property
- assign the chroma key RGB color to use to the [ImageOverlay\\_ChromaKeyRGBColor](#) property
- adjust the chroma key leeway % (optional) with the [ImageOverlay\\_ChromaKeyLeewayPercent](#) property (the default value is 25%)

E.g.:

```
VideoGrabber.PlayerFileName := 'my video clip.avi';
VideoGrabber.SetImageOverlayFromImageFile ('my background image.jpg');
VideoGrabber.ImageOverlay_ChromaKeyRGBColor := $FF0000; // the blue color
VideoGrabber.ImageOverlay_ChromaKeyLeewayPercent := 20; // 20%
VideoGrabber.ImageOverlay_ChromaKey := true;
VideoGrabber.ImageOverlayEnabled := true;
VideoGrabber.OpenPlayer;
```

To use several image overlay, see "**Applying several image overlays concurrently**" in the "[Graphic overlays](#)" chapter.

#### See Also

[ImageOverlay\\_ChromaKey](#) [ImageOverlay\\_ChromaKeyLeewayPercent](#)  
[ImageOverlay\\_ChromaKeyRGBColor](#)

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## Overlays before or after transforms

### Overlays before or after transforms

[Prev](#)

[Next](#)

Overlays before or after transforms

#### Description

The [OverlayBeforeTransform](#) property allows to specify if the graphic and text overlays must be performed before or after the transforms (like video rotation, cropping, etc...).

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## To retrieve a pixel value

### To retrieve a pixel value

[Prev](#)

[Next](#)



How to retrieve a pixel value

### Description

#### How to retrieve a pixel value

It is possible to retrieve the RGB value of a pixel at the (x, y) location in the video frame by invoking [GetRgBPixelAt](#) from the [OnFrameOverlayUsingDC](#) event.

Note: the video size is returned by the [VideoWidth](#) and [VideoHeight](#) functions.

---

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

---

## How to refresh the overlays while the video clip is paused

### How to refresh the overlays while the video clip is paused

[Prev](#)
[Next](#)

How to refresh the overlays while the video clip is paused

### Description

When the video clip is paused, the overlays are not refreshed by default, so the [OnFrameOverlayUsing...](#) events do not occur.

Invoke [RefreshPlayerOverlays](#) and TVideoGrabber will raise the [OnFrameOverlayUsing...](#) events and refresh the video frame with the updated overlays.

Look at the "free hand drawing" sample code in the [OnFrameOverlayUsingDC](#) event of the MainDemo project.

---

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

---

## Mouse events

### Mouse events

[Prev](#)
[Next](#)

Mouse events

### Description

The following mouse events are available when the mouse is moved over the video window:

**OnMouseDown:** [TOnVideoMouseUpDown](#)

**OnMouseMove:** [TOnVideoMouseMove](#)

**OnMouseUp:** [TOnVideoMouseUpDown](#)

**OnMouseWheel:** [TOnMouseWheel](#)  
(activated only if [MouseWheelEventEnabled](#) is set to true)

If [TranslateMouseCoordinates](#) is enabled (default), the mouse events return the real corresponding x, y positions on the video source, even if the video window is stretched.

---

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

---

## Player

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Encrypted, Password-Protected PDFs](#)

### Player features

#### Player features

[Prev](#)

[Next](#)

About the player.

#### Description

#### Player features

TVideoGrabber includes an advanced player that allows to play video clips, [capture video frames](#) and perform [frame overlay](#) when the [frame grabber](#) is enabled. It is possible to play multiple video clips seamlessly with the [Seamless playback](#) feature.

#### Opening / pausing / running / closing the clip

- set the [PlayerFileName](#) property with the full qualified path or the URL of the video clip,
- open the clip with [OpenPlayer](#)

When the clip is opened the [OnPlayerOpened](#) event occurs. From this event you can read the [PlayerDuration](#) and [PlayerFrameCount](#) properties, e.g. to setup a trackbar (see below the trackbar chapter)

*To open a clip in a paused state:*

```
VideoGrabber.PlayerFileName = "myclip.avi"
VideoGrabber.AutoStartPlayer = false
VideoGrabber.OpenPlayer\(\)
```

*To open and play a clip immediately:*

```
VideoGrabber.PlayerFileName = "myclip.avi"
VideoGrabber.AutoStartPlayer = true
VideoGrabber.OpenPlayer\(\)
```

*To pause a clip during playback:*

```
VideoGrabber.PausePlayer\(\)
```

*To resume the playback when the clip is paused:*

```
VideoGrabber.RunPlayer\(\)
```

*To close the clip:*

```
VideoGrabber.ClosePlayer\(\)
```

#### Player seeking

Note:

- if a video clip is seeking slow, invoke [MultipurposeEncoder\\_ReindexClip](#) to reindex it (*requires the optional Datastead Multipurpose Encoder to be installed*)
- to determine if a MP4 video clip needs a reindexing, invoke [MP4NeedsReindexing](#)

After opening the video clip, you can set a new position by assigning:

- either the [PlayerFramePosition](#) property, in the 1..[PlayerFrameCount](#) range
- either the [PlayerTimePosition](#) property, in the 0..[PlayerDuration](#) range (expressed in 100ns units, e.g. 2 seconds = 20000000)

*To set the current position at the frame #500:*

```
VideoGrabber.PlayerFramePosition = 500
```



To specify a new position located at "15 seconds":  
VideoGrabber.[PlayerTimePosition](#) = 150000000

### Frame counting and frame seeking

By default the frame counting / frame seeking starts from 1 (the 1st frame is the frame #1).

Alternatively it is possible to let TVideoGrabber start from the frame #0 by enabling the [FrameNumberStartsFromZero](#) property.

### Frame stepping

When the clip is paused, you can invoke [PlayerFrameStep](#) (1) to step to the next frame.

It is possible to specify a value > 1 to skip n-1 frames, e.g. [PlayerFrameStep](#) (10) will skip the next 9 frames and display the 10th frame.

*Note that with MPEG video clips [PlayerFramePosition](#) and [PlayerTimePosition](#) can reach only key frames, whereas [PlayerFrameStep](#) will step through all the frames (key frames **and** delta frames).*

### Seeking

If the clip is seekable it can be played backwards using [RunPlayerBackwards](#) , however the play back is not as smooth as normal play back.

*Note that you can get a more effective reverse playback by opening the clip through AVISynth, as explained in the [Advanced playback through AVISynth](#) chapter.*

To scan the clip quickly:

- forwards use [FastForwardPlayer](#) ,
- backwards use [RewindPlayer](#) ,

The scan speed can be modified using [PlayerFastSeekSpeedRatio](#) .

### Keeping the last video frame visible while opening the next video clip

By default after invoking [ClosePlayer](#), the video window is closed until the next clip is opened.

Enable the [VideoVisibleWhenStopped](#) property to keep the video window active after the previous video ends.

Then the last video frame will remain displayed until [OpenPlayer](#) is invoked and the new video clip starts playing.

### Playback boundaries

Instead of invoking [OpenPlayer](#):

- invoke [OpenPlayerAtFramePositions](#) to open a clip at the specified start frame
- invoke [OpenPlayerAtTimePositions](#) to open a clip at the specified stop frame

If the [KeepBound](#) parameter is true, the clip is played within the start and stop boundaries specified. When the clip is already opened, invoking [OpenPlayerAtFramePositions](#) or [OpenPlayerAtTimePositions](#) modifies the playback boundaries without reopening the clip.

### Playback progress

During playback, information about each video frame is reported by the [OnFrameProgress](#) event.

### End of stream

When the end of stream is reached the [OnPlayerEndOfStream](#) event occurs.

From this event you can invoke any task, e.g. you can invoke [RunPlayer](#) to restart the playback from the beginning.

### Paused player and CPU consumption

When a video clip is paused and:

- the [PlayerRefreshPausedDisplay](#) property is enabled,
- the [FrameGrabber](#) property is enabled,

the display is refreshed periodically at a frame rate specified by [PlayerRefreshPausedDisplayFrameRate](#),

in order to get the text and graphics overlays refreshed.

However this feature requires CPU. To reduce the CPU consumption when a video clip is paused, you can:

- specify a lower [PlayerRefreshPausedDisplayFrameRate](#) value,
- disable the [PlayerRefreshPausedDisplay](#) property.

### Playback speed

A clip is played at its normal speed when [PlayerSpeedRatio](#) = 1.

It is possible to play video clips at different speeds, however the playback speed range is larger if the audio rendering is disabled. To play a clip at a different speed:

- with audio rendering, enable [PlayerAudioRendering](#) and modify [PlayerSpeedRatio](#) in the 0.5 to 2.0 range,
- without audio rendering, disabled [PlayerAudioRendering](#) and modify [PlayerSpeedRatio](#) in a larger range.

### Playing clips at max speed without clock

It is possible to play the clips without clock, this guarantee e.g. that all the frames will be captured in burst mode (when [BurstMode](#) = true and [BurstCount](#) = 0).

To enable this feature, disable [UseClock](#) before invoking [OpenPlayer](#).

### Player trackbar

TVideoGrabber handles automatically a player trackbar (Delphi and C++Builder versions ONLY).

Simply put a trackbar on the form, and then associate it by assigning it from the Object Inspector to the [PlayerTrackBar](#) property.

See the [PlayerTrackBar](#) and the [PlayerTrackBarScale](#) properties for more information.

### Frame capture during playback

Video frames can be captured during playback. See the [Frame capture](#) chapter.

### Audio playback

For video clips that has an audio stream (when [IsPlayerAudioStreamAvailable](#) returns ts\_True), you can disable the rendering of the audio stream by setting [PlayerAudioRendering](#) to false. This can help to play video clips slower than 0.5 or faster than 2.

### Selecting the audio sound card

You can specify the soundcard used for the playback rendering by assigning the [AudioRenderer](#) index with the index of the sound card in the [AudioRenderers](#) list.

*e.g. if the index in the list is 2:*

VideoGrabber.AudioRenderer = 2

OR

*e.g. if the name in the list is "Speakers (Blackmagic Audio):"*

VideoGrabber.AudioRenderer = VideoGrabber.AudioRendererIndex ("Speakers (Blackmagic Audio)")

### Selecting only the right or left audio channel

It is possible to convert the right channel or the left channel as "mono" and to mute either the left, right or both channels, or to convert a mono channel into stereo with [AudioChannelRenderMode](#).

### Selecting one audio stream when the clip has multiple audio streams

When the clip has multiple audio streams it is possible to select only one audio stream with [AudioStreamNumber](#)

### Codec used during playback

It is possible to force the use of a given codec by assigning a codec name or a codec GUID to [PlayerForcedCodec](#) .

The video codec and audio codec using during playback are reported by the [PlayerVideoCodec](#) and [PlayerAudioCodec](#) properties.

### Specifying the buffering time when playing a streaming URL

The buffering time can be adjusted by invoking [ShowDialog](#) (dlg\_NetShowConfig)

### Authentication

If an authentication is required to connect to a streaming url, there are 2 ways to set an username and password required:

- either by invoking [SetAuthentication](#) (at\_StreamingUrl, "...username...", "...password..." ) before invoking OpenPlayer,
- either through the [OnAuthenticationNeeded](#) event that will occur when connecting if [SetAuthentication](#) has not been invoked yet.

### See Also

[Opening a clip or an IP URL from a background thread without blocking the main thread](#)  
[TOnPlayerStateChanged](#) [TOnThreadSync](#) [TPlayerState](#) [TThreadSyncPoint](#) [TOnPlayerBufferingData](#)  
[AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#)  
[AVIInfo2](#) [ClosePlayer](#) [EnableThreadMode](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#)  
[IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#)  
[OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OnThreadSync](#)  
[OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#)  
[PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#)  
[PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#)  
[PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#)  
[PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#)  
[RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#)  
[SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion

## Seamless playback

### Seamless playback

[Prev](#)

[Next](#)

Seamless playback

#### Description

The seamless playback feature allows to play a list of several clips **having the same format** as a single clip:

```
VideoGrabber.Playlist (pl_Clear, "");
VideoGrabber.Playlist (pl_Add, 'c:/folder/videoclip1.mp4');
VideoGrabber.Playlist (pl_Add, 'c:/folder/videoclip2.mp4');
VideoGrabber.Playlist (pl_Add, 'c:/folder/videoclip3.mp4');
VideoGrabber.PlayerFileName := 'PLAYLIST';
VideoGrabber.OpenPlayer;
More information can be found in the PlayList chapter.
```

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## Current player state

### Current player state

[Prev](#)

[Next](#)

Current player state

#### Description

#### Obtaining the current player state

At any time it is possible to know the current player state by reading the [PlayerState](#) property.

E.g., to know if the player is currently in normal playing state:

```
if PlayerState = ps_Playing then...
```

E.g. to know if the player is opened and playing normally, backwards or faster:

```
if PlayerState >= ps_Playing then...
```

E.g. to know if the player is paused:

```
if PlayerState = ps_Paused then...
```

### Player state notification

Each time the player state changes, the [OnPlayerStateChanged](#) event and returns the old and the new state (of [TPlayerState](#) type).

---

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation Process with HelpNDoc's Advanced Features](#)

---

## Trackbar

### Trackbar

[Prev](#)
[Next](#)

Implementation of a player trackbar

#### Description

#### Moving the position clip immediately while the trackbar position is moved

When [PlayerTrackBarSynchron](#) is enabled, moving the trackbar with the mouse **updates continuously** the position in the video clip, until the mouse button is released.

When [PlayerTrackBarSynchron](#) is disabled (**default**), the position is **updated only** when the mouse button is released.

This property requires the **full trackbar implementation** below (unless you are using the PlayerTrackbar property under Delphi or C++Builder, see remark below)

#### TRACKBAR IMPLEMENTATION

*Remark: if you are using Delphi or C++Builder, you can put a TTrackbar component on your form, assign it to the [PlayerTrackbar](#) property, it will be handled automatically so you can just ignore the implementations below.*

#### Minimal trackbar implementation

The minimal trackbar implementation consists to:

- place a trackbar on your form
- create a TVideoGrabber's [OnPlayerOpened](#) event, and set your trackbar max position by reading the [PlayerFrameCount](#) property from this event
- create a TVideoGrabber's [OnPlayerUpdateTrackbarPosition](#) event, and update your trackbar position with the FrameNumber parameter returned by this event
- create a trackbar's OnChange or OnValueChanged event, and update the [PlayerFramePosition](#) from this event with the current position of your trackbar

The problem with this minimal implementation is that you may notice a jerky trackbar behavior if you move the trackbar thumb while the clip is playing, because the trackbar position is upgraded periodically in the background by the [OnPlayerUpdateTrackbarPosition](#) event.

### FULL TRACKBAR IMPLEMENTATION IN VISUAL STUDIO

#### 1. put a trackbar component on your form

**2. create a TVideoGrabber's [OnPlayerOpened](#) event** that sets the min and max positions of the trackbar, e.g.:

```
private void axVideoGrabberNET1_OnPlayerOpened(object sender, System.EventArgs e)

tbrPlayer.Minimum = 0;
tbrPlayer.TickFrequency = 1;
tbrPlayer.Maximum = (int) axVideoGrabberNET1.PlayerFrameCount;
```

**3. create a TVideoGrabber's [OnPlayerUpdateTrackbarPosition](#) event** that updates your trackbar's position, e.g.:

```
private void axVideoGrabberNET1_OnPlayerUpdateTrackbarPosition(object sender,
Axvidgrab_NET.IVideoGrabberNETEvents_OnPlayerUpdateTrackbarPositionEvent e)

tbrPlayer.Value = (int) e.frameNumber;
```

**4. create a trackbar's [OnChange](#) or [OnValueChanged](#) event** that updates the TVideoGrabber's [PlayerFramePosition](#), e.g:

```
private void tbrPlayer_ValueChanged(object sender, System.EventArgs e)

axVideoGrabberNET1.PlayerFramePosition = tbrPlayer.Value;
```

**5. create a trackbar's [OnMouseDown](#), [OnMouseUp](#), [OnKeyDown](#) and [OnKeyUp](#) event** that invoke [NotifyPlayerTrackbarAction](#) with the corresponding TTrackbarAction parameter, e.g.:

```
private void tbrPlayer_MouseDown(object sender, System.Windows.Forms.MouseEventArgs e)

axVideoGrabberNET1.NotifyPlayerTrackbarAction (vidgrab_NET.TxTrackbarAction.tba_MouseDown);

private void tbrPlayer_MouseUp(object sender, System.Windows.Forms.MouseEventArgs e)

axVideoGrabberNET1.NotifyPlayerTrackbarAction (vidgrab_NET.TxTrackbarAction.tba_MouseUp);

private void tbrPlayer_KeyDown(object sender, System.Windows.Forms.KeyEventArgs e)

axVideoGrabberNET1.NotifyPlayerTrackbarAction (vidgrab_NET.TxTrackbarAction.tba_KeyDown);

private void tbrPlayer_KeyUp(object sender, System.Windows.Forms.KeyEventArgs e)

axVideoGrabberNET1.NotifyPlayerTrackbarAction (vidgrab_NET.TxTrackbarAction.tba_KeyUp);
```

Then run your app, the trackbar should work properly even when moving the thumb while the clip is playing.

---

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

---

## Playing static images and animated GIFS

### Playing static images and animated GIFS

[Prev](#)

[Next](#)

Playing static images  
**Description**

## Static images

Static images, like BMP and JPG files can be opened in the player.

They are opened in a paused state, so if [PlayerRefreshPausedDisplay](#) and [FrameGrabber](#) are enabled, it is possible to perform graphic processing and then recapture the frame.

The transform processings can also be applied (like [rotation](#), [brightness](#), [cropping](#), etc...)

## Animated GIFs

Animated GIFs can be opened exactly as video clips. The transform processings and graphic overlays can be applied (like [rotation](#), [brightness](#), [cropping](#), etc...).

---

Created with the Standard Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

## Playlist

### Playlist

[Prev](#)
[Next](#)

Using the playlist

### Description

### Overview

The playlist features lets you specify a list of video clips that have to be played automatically one after the other.

There are 2 ways to use the playlist:

**statically:** you specify a static set of clips, then play this set of clip like it was a single clip (the trackbar moves across the whole set of clips)

**dynamically:** you can dynamically add/remove clips while the playlist is playing by invoking `PlayList` (`pl_Add...`). In this case the trackbar is active for the current clip being played.

- all the playlist actions are performed by using the [Playlist](#) function, detailed below.
- the playlist content can be retrieved by the [GetPlaylist](#) function, that returns a list of strings separated by CR/LF
- the current index in the playlist is returned by the [PlaylistIndex](#) property (if opened dynamically)
- when the end of the playlist is reached, the [OnPlayerEndOfPlaylist](#) event occurs.
- it is possible to know if the playlist is currently playing by testing [IsPlaylistActive](#).

### The Playlist function

The playlist function accepts 2 parameters:

- a first parameter that specify the action
- a 2nd parameter that specify the video clip (used only for `pl_Add` and `pl_Remove`).

The first parameter can take one of the following values:

**pl\_Add:** adds the video clip (specified as 2nd parameter) to the playlist

**pl\_Remove:** removes the video clip (specified as 2nd parameter) from the playlist

**pl\_Clear:** clears the playlist

**pl\_Loop:** enables the loop mode (the playlist restart from the beginning when the end is reached)

**pl\_NoLoop:** disables the loop mode

**pl\_Play:** starts playing the playlist  
**pl\_Stop:** stops playing the playlist after the end of the current clip  
**pl\_Next:** starts playing the next clip (immediately)  
**pl\_Previous:** starts playing the previous clip (immediately)  
**pl\_SortAlpha:** sorts the playlist in the alphabetical order  
**pl\_SortRevAlpha:** sorts the playlist in the reverse alphabetical order  
**pl\_Random:** enable the "random" mode: the clips will be played in a random order.  
**pl\_Sequential:** disables the "random" mode: the clips will be played sequentially.  
**pl\_SpecifyPositions:** let specify a start position and/or a stop position

**Remark:** while the static playlist is playing, you can't control it with the playlist settings anymore, it behaves like a single video clip.

### Building the playlist

Simply invoke Playlist (pl\_Add, "... path of the video clip...") as many times as required. Pass an empty string for the "VideoClip" parameter when it is not required (it is required only for pl\_Add and pl\_Remove).  
 E.g.:

```
videograbber.Playlist (pl_Clear, '')
videograbber.Playlist (pl_Add, 'vg000004.avi')
videograbber.Playlist (pl_Add, 'vg000002.avi')
videograbber.Playlist (pl_Add, 'StaticImage1.BMP')
videograbber.Playlist (pl_Add, 'vg000003.avi')
videograbber.Playlist (pl_Add, 'vg000005.avi')
videograbber.Playlist (pl_Add, 'vg000001.avi')
videograbber.Playlist (pl_Add, 'StaticImage2.JPG')
```

### Starting the dynamic playlist

Be sure to set all the required options (like pl\_Loop, pl\_SortAlpha or pl\_Random), then invoke pl\_Play. E.g.:

```
videograbber.Playlist (pl_Clear, '')
videograbber.Playlist (pl_Add, 'vg000004.avi')
videograbber.Playlist (pl_Add, 'vg000002.avi')
videograbber.PlayerDuration = 5000000 // specifies 5 seconds of display time for StaticImage1.BMP
videograbber.Playlist (pl_Add, 'StaticImage1.BMP');
videograbber.Playlist (pl_Add, 'vg000003.avi')
videograbber.Playlist (pl_Add, 'vg000005.avi')
videograbber.Playlist (pl_Add, 'vg000001.avi')
videograbber.PlayerDuration = 8000000 // specifies 8 seconds of display time for StaticImage2.JPG
videograbber.Playlist (pl_Add, 'StaticImage2.JPG')
videograbber.PlaylistIndex := 0
videograbber.Playlist (pl_SortAlpha, '')
videograbber.Playlist (pl_Loop, '')
videograbber.Playlist (pl_Random, '')
videograbber.Playlist (pl_Play, '');
```

### Starting the static playlist

```
videograbber.Playlist (pl_Clear, '');
videograbber.Playlist (pl_Add, 'vg000004.avi');
videograbber.Playlist (pl_Add, 'vg000002.avi');
videograbber.Playlist (pl_Add, 'vg000003.avi');
videograbber.Playlist (pl_Add, 'vg000005.avi');
videograbber.Playlist (pl_Add, 'vg000001.avi');
videograbber.PlayerFileName = 'PLAYLIST'
videograbber.OpenPlayer()
```

### Specifying a custom video size and/or frame rate to the static playlist



By default the static playlist plays all the clips at the same video size (640x480) and at the same frame rate (29.97 fps)

- to customize the frame rate, set a [FrameRate](#) value > 0
- to customize the video size, invoke [UseNearestVideoSize](#) (width, height, false) before invoking OpenPlayer

E.g.:

```
...
VideoGrabber.PlayList (pl_Add, "vg0000006.avi")
VideoGrabber.PlayerFileName = 'PLAYLIST'
VideoGrabber.FrameRate = 25.0
VideoGrabber.UseNearestVideoSize (720, 480, false)
VideoGrabber.OpenPlayerPlayer()
```

You can later reset the video size by invoking:

```
UseNearestVideoSize (0, 0, false)
```

and the frame rate by setting:

```
FrameRate = 0
```

## Pausing the playlist

Simply invoke [PausePlayer](#).

## Stopping the playlist

- to stop the playlist at the end of the current clip, invoke Playlist (pl\_Stop)
- to close the playlist immediately, invoke [ClosePlayer](#)

## specifying start/stop positions for each clip

Invoke VideoGrabber.PlayList (pl\_SpecifyPositions,...) and immediately after [OpenPlayerAtTimePositions](#) or [OpenPlayerAtFramePositions](#) (this will not open the clips but just register the positions)

```
VideoGrabber.PlayList (pl_Add, 'myvideoclip1.wmv');
VideoGrabber.PlayList (pl_SpecifyPositions, 'myvideoclip1.wmv');
VideoGrabber.OpenPlayerAtTimePositions (50000000, 120000000, true, true);

VideoGrabber.PlayList (pl_Add, 'myvideoclip2.wmv');
VideoGrabber.PlayList (pl_SpecifyPositions, 'myvideoclip2.wmv');
VideoGrabber.OpenPlayerTimePositions (30000000, 130000000, true, true);

VideoGrabber.PlayerFileName := 'PLAYLIST';
VideoGrabber.Openplayer();
```

In this example the first clip will play 7 seconds (from 7 seconds to 12 seconds) and then the second will play 10 seconds (from 3 seconds to 13 seconds)  
to let the clip play until the end specify -1 as stop position)

## Using static images or animaged GIFs in the playlist

Static images (like BMP, JPG files) can be opened alternatively with video clips in the dynamic playlists. To specify the display time for static images, assign the [PlayerDuration](#) with the desired value **BEFORE** the PlayList (pl\_Add...) statement (the display time is expressed as 100ns units, e.g. 7 seconds = 70000000).

E.g.:

```
VideoGrabber.PlayerDuration = 30000000; // will set 3 seconds for myvideoclip1
VideoGrabber.PlayList (pl_Add, 'myvideoclip1.wmv');
```



```
VideoGrabber.PlayerDuration = 40000000; // will set 4 seconds for myvideoclip2
VideoGrabber.PlayList (pl_Add, 'myvideoclip2.wmv');
...
VideoGrabber.PlayList (pl_Play, "");
```

### See Also

[TPlaylist](#) [Video formats](#) [GetPlaylist](#) [IsPlaylistActive](#) [OnPlayerEndOfPlaylist](#) [Playlist](#) [PlaylistIndex](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

## Playing a clip from a TStream

### Playing a clip from a TStream

[Prev](#)
[Next](#)

Playing a clip from a TStream.

#### Description

It is possible to play in the media player a video clip stored in a TMemoryStream, TFileStream or TStream descendent.

Simply assign your TStream descendent to the [SourceStream](#) property. When this property is assigned **before** invoking [OpenPlayer](#), the stream data is used as source by the player.

To go back to the "normal player behavior", set a **nil** (or NULL in C++) value to [SourceStream](#) **after** invoking [ClosePlayer](#), to let TVideoGrabber use for the next [OpenPlayer](#) the clip specified in the [PlayerFileName](#) property.

Created with the Standard Edition of HelpNDoc: [HelpNDoc's Project Analyzer: Incredible documentation assistant](#)

## StartPreview or StartRecording from video clips or URLs

### StartPreview or StartRecording from video clips or URLs

[Prev](#)
[Next](#)

Previewing or recording video clips

#### Description

#### Previewing or recording video clips

It is possible to preview or record a video clip (or a streaming URL) by invoking StartPreview or StartRecording as follows:

- set VideoSource = vs\_VideoFileOrURL
- set VideoSource\_FileOrUrl = "http://.....YourStreamingUrl..." or ("mms://...yourStreamingUrl....")
- (optionally) invoke SetAuthentication (ap\_StreamingUrl, "...username...", "...password...")
- then invoke StartPreview or StartRecording

#### Authentication

If an authentication is required to connecto a streaming url, there are 2 ways to set an username and password required:

- either by invoking [SetAuthentication](#) (at\_StreamingUrl, "...username...", "...password..." ) before invoking OpenPlayer,

- either through the [OnAuthenticationNeeded](#) event that will occur when connecting if [SetAuthentication](#) has not been invoked yet.

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## AVI / ASF information and header attributes

### AVI / ASF information and header attributes

[Prev](#)

[Next](#)

AVI / ASF information and header attributes of a video clip

#### Description

#### AVI information of a video clip

The [AVIInfo](#) function retrieves the following information:

- duration,
- frame count,
- video width
- video height,
- video frame rate in frames per second,
- audio bit rate in Hertz,
- name of the video codec,
- name of the audio codec.

This function requires the parameters to be passed by reference.

For languages that pass parameters only by value (e.g. JavaScript), use [AVIInfo2](#) instead.

#### AVI / ASF header attributes of a video clip

Invoke [AVIHeaderInfo](#) to retrieve the attributes of an existing video clip, if any.

Created with the Standard Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

## Playing pictures

### Playing pictures

[Prev](#)

[Next](#)

Playing pictures

#### Description

The player can open a single picture, e.g.:

```
VideoGrabber.PlayerFileName = "mypicture.jpg"
VideoGrabber.OpenPlayer()
```

However you may need a picture to be displayed at a given frame rate, e.g. to perform graphics or text overlays that may change while the picture is displayed.

In this case proceed rather this way:

- set [VideoSource](#) = vs\_JPEGsOrBitmaps,
- create a [OnVideoFromBitmapsNextFrameNeeded](#) event and pass the file name of your .JPG to the BMPorJPEGFile parameter of this event,
- set [FrameRate](#) with the frame rate needed
- then invoke [StartPreview](#)

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## Synchronization of several player components

### Synchronization of several player components

[Prev](#)
[Next](#)

Synchronization of several player components

#### Description

#### Synchronization of several player components

It is possible to synchronize several player components.

One component is declared as the master component, the other components are the slave components.

Example:

```
VideoGrabber2.Synchronized = true
VideoGrabber2.SynchronizationRole = sr_Slave
```

```
VideoGrabber3.Synchronized = true
VideoGrabber3.SynchronizationRole = sr_Slave
```

```
VideoGrabber1.Synchronized = true
VideoGrabber1.SynchronizationRole = sr_Master
VideoGrabber1.PlayerFileName = "myclip.avi"
VideoGrabber1.AutoStartPlayer = false
VideoGrabber1.OpenPlayer()
VideoGrabber1.StartSynchronized\(\)
```

This will open the clip in the 3 components at the same time. Starting from now all the player actions below performed on VideoGrabber1 are applied at the same time on VideoGrabber2 and VideoGrabber3:  
by invoking the following functions:

[OpenPlayer](#)  
[ClosePlayer](#)  
[RunPlayerBackwards](#)  
[PausePlayer](#)  
[StopPlayer](#)  
[RunPlayer](#)  
[RewindPlayer](#)  
[FastForwardPlayer](#)

by assigning the following properties:

[PlayerTimePosition](#)  
[PlayerFramePosition](#)

#### Shifting the synchronization position of one slave player

You can set a delta position on a slave player (after the clip is opened on all the components) as follows:

- disable temporarily the [Synchronized](#) property on this slave
- set a new position on the slave (with [PlayerFramePosition](#) or [PlayerTimePosition](#))
- enable the [Synchronized](#) property on this slave

Now all position changes on the master component will be performed on the slave by applying the delta position set while the Synchronized property was disabled.

Look at the *SynchronizedPlayers demo project* included in the package for sample code.

## Video 360°

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

### Decoding of 360° videos

#### Decoding of 360° videos

[Prev](#)
[Next](#)

#### 360 Videos decoding

##### Description

The decoding of 360 videos requires the bundle of the TVideoGrabber SDK + Datastead RTSP/RTMP/HTTP/ONVIF Source Filter (v7.8.2.0 or higher).

To enable the decoding of 360° videos, enable the [v360\\_Enabled](#) property.

When [v360\\_MouseAction](#) is activated, the mouse motion controls the point of view, and the mouse wheel controls the angles (zoom), either when the mouse button is down, either when the mouse button is released, depending on the [Tv360\\_MouseAction](#) value.

When moving the mouse, the "quantity of displacement" is controlled by [v360\\_MouseActionPercent](#) (10% by default).

The mouse wheel controls the zoom-in or zoom-out.

To return to the default angles, invoke [v360\\_ResetAnglesToDefault](#)

The point of view can be controled by [v360\\_SetYawPitchRoll](#) or [v360\\_AddYawPitchRoll](#).

The current point of view is returned by [v360\\_GetYawPitchRoll](#)

The zoom can be controled directly by invoking [v360\\_SetAngle](#) (v360\_fov\_Horizontal, Value) or [v360\\_SetAngle](#) (v360\_fov\_Vertical, Value).

The master angle is the angle from which is calculated the opposite angle to maintain the aspect ratio, by default it is v360\_fov\_Horizontal but can be changed to v360\_fov\_Vertical by modifying the [v360\\_MasterAngle](#) property.

The aspect ratio is controled automatically when [v360\\_AspectRatio](#) = -1 (default).

This can be deactivated by setting a value > 0.0 (e.g. set 1.777778 to force a 16:9 aspect ratio, or 2.0 to set a 2:1 aspect ratio, etc...).

**Supported stereo formats:** ([v360\\_SetStereoFormat](#))

sf\_2DMono,  
sf\_SideBySide,  
sf\_TopBottom

**Supported projections:** ([v360\\_SetProjection](#))

ipp\_Equirectangular,  
ipp\_Cubemap\_3\_2,  
ipp\_Cubemap\_6\_1,  
ipp\_Equiangular,  
ipp\_Flat,  
ipp\_Dual\_fisheye,  
ipp\_Barrel,  
ipp\_Cubemap\_1\_6,  
ipp\_Stereographic,  
ipp\_Mercator,  
ipp\_Ball,  
ipp\_Hammer,  
ipp\_Sinusoidal,  
ipp\_Fisheye,  
ipp\_Pannini,  
ipp\_Cylindrical,  
ipp\_Perspective,  
ipp\_Tetrahedron,  
ipp\_Barrel\_split,  
ipp\_Tsypyrmaid,  
ipp\_Hequirectangular,  
ipp\_Equisolid,  
ipp\_Orthographic,

ipp\_Octahedron

**Supported interpolations:** ([v360\\_SetInterpolation](#))

ipl\_Bilinear,  
ipl\_Nearest,  
ipl\_Lagrange9,  
ipl\_Bicubic,  
ipl\_Lanczos,  
ipl\_Spline16,  
ipl\_Gaussian,  
ipl\_Mitchell

---

Created with the Standard Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

## PIP

---

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

---

### PIP (Picture In Picture)

#### PIP (Picture In Picture)

[Prev](#)

[Next](#)

PIP (Picture In Picture)

#### Description

#### PIP (Picture In Picture) between several TVideoGrabber components

During preview, recording or playback, a TVideoGrabber component can receive and display the video from others TVideoGrabber components, and display their videos as rectangles in "PIP" mode (Picture in Picture)

This is set up by invoking [Mixer\\_SetupPIPFromSource](#) and passing as parameter the TVideoGrabber component used as source, the source rectangle that should be captured, and the destination rectangle where the PIP should be displayed.

The first boolean parameter specifies if the PIP must be activated or deactivated

The last boolean parameter specifies if the PIP must be located at the top (useful only if several PIPs are overlapped)

It is possible to make several PIPs on the same video frame.

E.g. let's suppose the whole VideoGrabber2 video frame should be displayed as a 80x60 rectangle at the x=10, y=10 location in VideoGrabber1, invoke:

```
VideoGrabber1.Mixer_SetupPIPFromSource (VideoGrabber2.UniqueID, 0, 0, 0, 0, true, 10, 10,
```

E.g. let's suppose a 160x120 rectangle of VideoGrabber2 located at x=40 y=60, should be displayed as a 80x60 rectangle at the 10, 10 location in VideoGrabber1, invoke:

```
VideoGrabber1.Mixer_SetupPIPFromSource (VideoGrabber2.UniqueID, 40, 60, 160, 120, true, 1
```

You can re-invoke this function on the fly several times, e.g. to activate/deactivate the PIP or to modify its location or size.

---

Created with the Standard Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

---

## Streaming

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

### Streaming through the Datstead Encoder

#### Streaming through the Datstead Encoder

[Prev](#)

[Next](#)

Streaming through the Datstead Encoder

##### Declaration

```
procedure TfrmMainForm.btnMultipurposeUDPStreamingClick(Sender: TObject);
```

```
begin
```

```
  if FMultipurposeUDPStreamingID=-1 then begin
```

```
    FMultipurposeUDPStreamingID := VideoGrabber.Encoders_CreateInstanceForStreaming  
    ('udp://239.255.0.1:10124');
```

```
  end;
```

```
  if FMultipurposeUDPStreamingID > -1 then begin
```

```
    /// uncomment one of the lines below to use a specific codec, e.g.:
```

```
    //VideoGrabber.Encoder_SetStr (FMultipurposeUDPStreamingID, Enc_Video_Codec, 'hevc');
```

```
    //VideoGrabber.Encoder_SetStr (FMultipurposeUDPStreamingID, Enc_Video_Codec, 'h264');
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Video_Enabled_bool, 1);
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Audio_Enabled_bool, 1);
```

```
    if not chkStreamingVideoEnabled.Checked then begin
```

```
      VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Video_Enabled_bool, 0);
```

```
    end;
```

```
    if not chkStreamingAudioEnabled.Checked then begin
```

```
      VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Audio_Enabled_bool, 0);
```

```
    end;
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Video_BitRate_kb, 2000);
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Video_rc_MinBitRate_kb, 1800);
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Video_rc_MaxBitRate_kb, 2200);
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_Video_IDR_Interval, 30);
```

```
    /// uncomment to enable the GPU encoder, if available for h264 or hevc
```

```
    // VideoGrabber.Encoder_SetInt (FStreamingEncoderId, Enc_Video_GPU_Encoder, LongInt
```

```
    (Enc_GPU_Auto)); // or specify directly Enc_GPU_Intel_QSV, Enc_GPU_NVidia_NVENC or
```

```
    Enc_GPU_AMD_AMF
```

```
    VideoGrabber.Encoder_SetInt (FMultipurposeUDPStreamingID, Enc_IsActive_bool, 1);
```

```
  end;
```

```
  VideoGrabber.VideoSource := vs_ScreenRecording;
```

```
  VideoGrabber.UseNearestVideoSize (1920, 1080, true);
```

```
  VideoGrabber.StartPreview();
```

```
end;
```

```
private void btnMultipurposeUDPStreaming_Click(object sender, EventArgs e)
```

```
if (m_MultipurposeUDPStreamingID == -1)
```

```
  m_MultipurposeUDPStreamingID=VideoGrabber.Encoders_CreateInstanceForStreaming("udp://239.255.0.  
1:10124");
```

```
if (m_MultipurposeUDPStreamingID > -1)
```

```
  /// uncomment one of the lines below to use a specific codec, e.g.:
```

```
  //VideoGrabber.Encoder_SetStr (FMultipurposeUDPStreamingID, TEncoder_str.Enc_Video_Codec,  
  "hevc");
```

```
//VideoGrabber.Encoder_SetStr (FMultipurposeUDPStreamingID, TEncoder_str.Enc_Video_Codec,
"h264");

VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_Video_Enabled, 1);
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_Audio_Enabled, 1);
if (!chkVideoStreamingEnabled.Checked)
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_Video_Enabled, 0);

if (!chkAudioStreamingEnabled.Checked)
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_Audio_Enabled, 0);

VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_Video_BitRate_kb,
2000);
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID,
TEncoder_int.Enc_Video_rc_MinBitRate_kb, 1800);
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID,
TEncoder_int.Enc_Video_rc_MaxBitRate_kb, 2200);
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_Video_IDR_Interval,
30);
/// uncomment to enable the GPU encoder, if (available for h264 or hevc
// VideoGrabber.Encoder_SetInt (FStreamingEncoderId, TEncoder_int.Enc_Video_GPU_Encoder, LongInt
(TEncoder_int.Enc_GPU_Auto)); // or specify directly Enc_GPU_Intel_QSV, Enc_GPU_NVidia_NVENC or
Enc_GPU_AMD_AMF
VideoGrabber.Encoder_SetInt(m_MultipurposeUDPStreamingID, TEncoder_int.Enc_IsActive_bool, 1);

VideoGrabber.VideoSource=vs_ScreenRecording;
VideoGrabber.UseNearestVideoSize (1920, 1080, true);
VideoGrabber.StartPreview();
```

## Description

The installation and functions overview are described in the [Datastead Encoder](#) chapter.

## See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#)  
[OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

---

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)

---

## MMS streaming

### MMS streaming

[Prev](#)

[Next](#)

## MMS Streaming

### Description

The MMS streaming is activated when [NetworkStreaming](#) = ns\_ASFDirectNetworkStreaming.

The TCP port used can be specified with [ASFNetworkPort](#).

If [ASFNetworkPort](#) = 0, an available port is automatically assigned. This port is returned by the [OnDirectNetworkStreamingHostUrl](#). This event returns also the **full URL** that can be given to the users to connect to this network stream.

The streaming URL that must be given to users to connect to the current stream is also returned by the [StreamingURL](#) property.

### Increasing the default limitation of 5 concurrent clients

By default a standard ("non-server") Windows PC is limited to 5 concurrent connections. This limit can be increased to a maximum of 50 connections by modifying the value of the HKEY\_CLASSES\_ROOTMedia ToolsMaxClientConnections registry key.

### Controlling the users connected

- the maximum number of users that can connect to this network stream is specified by the [ASFNetworkMaxUsers](#) property (default value : 5 users).
- the current number of users connected is returned by the [ASFStreaming\\_GetConnectedClientsCount](#) function.
- the list of the users connected is returned by the [ASFStreaming\\_GetConnectedClients](#) string (as lines separated by CR/LF)
- it is possible to block an IP address (or a group of IP addresses) by invoking [ASFStreaming\\_SetAuthorization](#)
- the list of the blocked IP addresses is returned by the [ASFStreaming\\_GetAuthorizationList](#) string (as lines separated by CR/LF)
- to clear the list of the blocked IP addresses, invoke [ASFStreaming\\_ResetAuthorizations](#).

### SAMPLE CODE (server side)

Streaming a live video source (video only):

```
VideoGrabber1.VideoDevice := 0;
VideoGrabber1.NetworkStreaming := ns_ASFDirectNetworkStreaming;
VideoGrabber1.NetworkStreamingType := nst_VideoStreaming;
VideoGrabber1.ASFNetworkPort := 10500;
VideoGrabber1.StartPreview;
```

Streaming a live video source (audio + video):

```
VideoGrabber1.VideoDevice := 0;
VideoGrabber1.AudioDevice := 0;
VideoGrabber1.NetworkStreaming := ns_ASFDirectNetworkStreaming;
VideoGrabber1.NetworkStreamingType := nst_AudioVideoStreaming;
VideoGrabber1.ASFNetworkPort := 10500;
VideoGrabber1.StartPreview;
```

Streaming a video clip:

```
VideoGrabber1.VideoSource := vs_VideoFileOrURL;
VideoGrabber1.VideoSource_FileOrUrl := " replace by the file path to your video clip"
VideoGrabber1.NetworkStreaming := ns_ASFDirectNetworkStreaming;
VideoGrabber1.NetworkStreamingType := nst_AudioVideoStreaming;
VideoGrabber1.ASFNetworkPort := 10500;
VideoGrabber1.StartPreview;
```

To view the streaming on the client, you need to open the server URL, that is built as follows:

**mms:// IP : port**

E.g. if the server IP is 192.168.0.123 and the streaming port is 10500 the URL will be mms://192.168.0.123:10500

To know the IP address of the server.

- either run IPConfig on the server to show it,
- either use the OnDirectNetworkStreamingHostUrl event (on the server) that returns directly the URL to open in the client when starting the preview or recording (HostUrl parameter).

### SAMPLE CODE (client side)



*E.g. if the server IP is 192.168.0.123 and the port is 10500 the URL will be mms://192.168.0.123:10500*

- Without the optional [Datastead RTSP DirectShow source filter](#):

```
VideoGrabber1.VideoSource := vs_VideoFileOrURL;
VideoGrabber1.VideoSource_FileOrUrl := 'mms://192.168.0.123:10500';
VideoGrabber1.StartPreview;
```

- With the optional [Datastead RTSP DirectShow source filter](#) installed on the client:

```
VideoGrabber1.VideoSource := vs_VideoFileOrURL;
VideoGrabber1.VideoSource_FileOrUrl := 'mmsh://192.168.0.123:10500';
VideoGrabber1.StartPreview;
```

---

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

---

## Streaming using Newtek NDI

### Streaming using Newtek NDI

[Prev](#)
[Next](#)

Streaming using NEWTEK NDI

#### Description

The NDI streaming requires the Newtek NDI runtime, and the optional Datastead NDI Filters package that can be downloaded here :

<https://www.datastead.com/downloads/>

The installation of the Datastead NDI filters and the Newtek NDI runtime are described in the INSTALL\_README.rtf

#### Sample code to send a stream to the network:

```
VideoGrabber.VideoSource = ...
VideoGrabber.NDIName = "MYSTREAM";
VideoGrabber.NetworkStreaming = ns_NDI;
VideoGrabber.NetworkStreamingType = nst_VideoStreaming; // or nst_AudioVideoStreaming if audio
needed
VideoGrabber.StartPreview();
E.g. if the video source is a Logitech 930e webcam:
VideoGrabber.VideoSource = vs_VideoCaptureDevice;
VideoGrabber.VideoDevice = VideoGrabber.VideoDeviceIndex ("Logitech Webcam C930e");
VideoGrabber.AudioDevice = VideoGrabber.AudioDeviceIndex ("Microphone (Logitech Webcam C930e)");
VideoGrabber.NDIName = "MYSTREAM";
VideoGrabber.NetworkStreaming = ns_NDI;
VideoGrabber.NetworkStreamingType = nst_AudioVideoStreaming;
VideoGrabber.StartPreview();
```

#### Sample code to play a stream from the network:

```
VideoGrabber.VideoSource = vs_VideoFileOrURL;
VideoGrabber.VideoSource_FileOrURL = "ndi://DESKTOP-5B66SBT (MYSTREAM)";
VideoGrabber.AudioDeviceRendering = true ;
VideoGrabber.StartPreview();
```

#### Notes:

- replace "DESKTOP-5B66SBT" by the name of the computer that is streaming to the network, and "MYSTREAM" by the real stream name to use.
- if the sender and receiver are on the same computer, the computer name can be replaced by "localhost", e.g.:

```
VideoGrabber.VideoSource_FileOrURL = "ndi://localhost (MYSTREAM)";
```

#### See Also

[TNDIFormatType](#) [NDIFormatType](#) [NDIName](#) [NetworkStreaming](#)

## Video window - Display - Aspect Ratio - Dual display

### Video window

#### Video window

[Prev](#)[Next](#)

Video size overview.

#### Description

#### Video render

You can choose the video renderer through the [VideoRenderer](#) property:

**vr\_AutoSelect:** selected automatically

**vr\_VMR9:** video renderer 9 (VMR9) -> good quality, more CPU. It buffers 3 frames, this can cause positioning problems in the player.

**vr\_VMR7:** video renderer 7 (VMR7) -> good quality, less CPU

**vr\_EVR:** enhanced video renderer (EVR), best quality, more CPU

**vr\_StandardRenderer:** standard renderer -> average quality, less CPU

**vr\_OverlayRenderer:** overlay renderer -> good quality, low CPU

**vr\_RecordingPriority:** recommended if the quality of the recording is more important than the quality of the preview.

**vr\_None:** no renderer

#### Keeping the last video frame visible while opening the next video clip

By default after invoking [StopPreview](#), [StopRecording](#) or [ClosePlayer](#), the video window is closed until the next clip is opened.

Enable the [VideoVisibleWhenStopped](#) property to keep the video window active and the last frame displayed until [StartPreview](#), [StartRecording](#) or [OpenPlayer](#) start displaying video frames.

#### Show / hide the video window

Enable [Display\\_Active](#) to specify if the video window must be constructed when invoking StartPreview / StartRecording / OpenPlayer.

Then the video window can be shown or hidden when the preview/recording/playback is running by enabling or disabling [Display\\_Visible](#).

#### Size of the video window & auto resizing

The control is automatically resized by TVideoGrabber when [Display\\_AutoSize](#) is enabled, otherwise it depends of the Width and Height properties.

When [Display\\_AutoSize](#) is enabled:

- selecting a [VideoSize](#) in the [VideoSizes](#) list resizes the control to the selected size,

remark: the size of the control can be stretched to a given percent of the native video size by using [PreviewZoomSize](#). This property does not affect the frame capture size and the recording size.

#### Display aspect ratio

Video clips can have a display aspect ratio different than the frame aspect ratio. TVideoGrabber detects the display aspect ratio and takes it in account when the [AdjustPixelAspectRatio](#) property is enabled (see below).

In this case, the [VideoWidth\\_PREFERREDAspectRatio](#) and the [VideoHeight\\_PREFERREDAspectRatio](#) properties return the preferred width and height that should be used to display the clip. These properties are useful if you want to control yourself the video size when [Display\\_AutoSize](#) is disabled.

To display the video in its correct display aspect ratio:

- if you want the control to be resized automatically, set [AdjustPixelAspectRatio](#) = true **and** [Display\\_AutoSize](#) = true

- if you want to specify yourself the size of the control, set [AdjustPixelAspectRatio](#) = true **and** [Display\\_AutoSize](#) = false, and read the [VideoWidth](#) [PreferredAspectRatio](#) and [VideoHeight](#) properties from the [OnPlayerOpened](#) event to get the correct video width and height of the video clip.

To display the video AS IS:

- if you want the control to be resized automatically, set [AdjustPixelAspectRatio](#) = false **and** [Display\\_AutoSize](#) = true

- if you want to specify yourself the size of the control, set [AdjustPixelAspectRatio](#) = false **and** [Display\\_AutoSize](#) = false **and** [Display\\_AspectRatio](#) = ar\_NoResize and read the [VideoWidth](#) and [VideoHeight](#) properties from the [OnPlayerOpened](#), [OnPreviewStarted](#) or [OnRecordingStarted](#) events.

### Location of the video frames within the video window

When [Display\\_AutoSize](#) = false, the aspect ratio is controlled by the [Display\\_AspectRatio](#) property. The possible settings are:

**ar\_Box**: the video appears in full within the control in a letterbox (bars on the top and the bottom) or pillarbox (bars on the left and the right)

**ar\_NoResize**: the video appears in its "native" size. If the native size is larger than then control, it appears centered within the control. If the native size is larger than the control, the video appears truncated.

**ar\_Stretch**: the video is stretched as needed to fill the control

**ar\_PanScan**: from pan to tilt, according to the [Display\\_PanScanRatio](#) value (in the 0..100 range, default value 50).

### Zooming the frame capture

The size of captured frames can be stretched to a given percent of the native video size by using [FrameCaptureZoomSize](#) .

### Analog video sources

For non-DV video sources, [VideoSizes](#) returns the list of video sizes really supported by the [current video capture device](#) (e.g. 320x240, 352x288, 640x480, etc...)/

### DV video sources

DV video sizes are standard.

When the current video capture device is a DV source, the [VideoSizes](#) returns "dc", "quarter", "half" and "full". The resulting video size depends of the video signal (PAL or NTSC):

<a href="#">VideoSizes</a> list	NTSC	PAL
default	360x240	360x288
dc	88x66	88x72
quarter	180x120	180x144
half	360x240	360x288
full	720x480	720x576

### Recording size

The recording size depends of the [VideoSize](#) property, except when [UseNearestVideoSize](#) is used.

However, when capturing DV video sources and [RecordingInNativeFormat](#) is enabled, the video stream is always saved to AVI in "full" size, and [VideoSize](#) and [UseNearestVideoSize](#) apply in this case only to the preview control size.

### refreshing the video sizes

Any listbox that uses the video sizes of the current video capture device should be refreshed with [VideoSizes](#) ([VideoSize](#) is the current index in this list) when the [OnVideoDeviceSelected](#) event occurs. See the *MainDemo* project for sample code.

### Video quality under Windows Vista / Windows 7 and later

It is possible to improve the quality of the video display (e.g. in full screen mode) when playing MPEG clips and the frame grabber is disabled, see the [Aero](#) property.

### Custom renderer

If your video display board is shipped with a custom renderer, you can tell TVideoGrabber to use it. You have to invoke the `ThirdPartyFilter_AddToList` function before invoking `StartPreview`, and pass the CLSID of your third-party filter.

E.g.:

```
...
uses ComObj;
...
procedure TfrmMainForm.Button1Click(Sender: TObject);
const
  CLSID_MyRendererFilter: TGUID = '70e102b0-5556-11ce-97c0-00aa0055595a';
begin
  if VideoGrabber.ThirdPartyFilter_AddToList (tpf_VideoRenderer, GUIDToString (CLSID_MyRendererFilter)) then
    ShowMessage ('filter added');
    VideoGrabber.StartPreview;
  end;
end;
```

Feel free to contact us at [support@datastead.com](mailto:support@datastead.com) if you don't know how to retrieve the CLSID of your renderer filter.

### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

---

Created with the Standard Edition of [HelpNDoc: Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

---

## Dual display

### Dual display

[Prev](#)
[Next](#)

Dual display.

#### Description

##### overview

It is possible to display 2 identical video windows, e.g. for a teacher to see the video preview on a 1st monitor, and for students to see the same video on a 2nd monitor in full screen mode.

The default video window is handled through the "Display\_" properties and methods.

The 2nd video window is handled through the "DualDisplay\_" identical properties and methods.

##### default video window

By default, only the 1st video window is active ([Display\\_Active](#) = true) and the 2nd video window is disabled ([DualDisplay\\_Active](#) = false).

By default, the 1st video window is embedded in the TVideoGrabber control ([Display\\_Embedded](#) = true).

##### activation/deactivation of the dual display

To activate the 2nd video window, enable [DualDisplay\\_Active](#) before calling [StartPreview](#) , [StartRecording](#) or [OpenPlayer](#) .

Then the 2nd video window can be shown or hidden when the preview/recording/playback is running by enabling or disabling [DualDisplay\\_Visible](#).

##### multi-monitors display

You can send each video window on the monitor of your choice. Simply assign the [Display\\_Monitor](#) or [DualDisplay\\_Monitor](#) property with the index of the desired monitor.

2 global functions help you to get information on the available monitors:

1) [MonitorsCount](#) returns the number of monitors available on the current platform.

Therefore [Display\\_Monitor](#) or [DualDisplay\\_Monitor](#) accept a value in the 0..[MonitorsCount](#) -1 range.  
**2) [MonitorBounds](#)** returns the bounds of the specified monitor in the 0..[MonitorsCount](#) -1 range.  
 The bounds returned are intended to help you to position the video window within the screen bounds.

### window location

When it is not embedded ([Display\\_Embedded](#) = false), the video window:

- can be placed on the screen by assigning [Display\\_Left](#) and [Display\\_Top](#) ,
- can be sized by assigning [Display\\_Width](#) and [Display\\_Height](#) .

However, rather than assigning these properties one by one, you can set all of them at the same time by using [Display\\_SetLocation](#) .

### See Also

[Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_TransparentColorEnabled](#)  
[Display\\_TransparentColorValue](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#)  
[DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#)  
[DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#)  
[DualDisplay\\_PanScanRatio](#) [DualDisplay\\_SetLocation](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#)  
[DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#)  
[DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#)  
[DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#)  
[SetWindowTransparency](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#)

---

Created with the Standard Edition of HelpNDoc: Transform Your Documentation Workflow with HelpNDoc's Intuitive UI

---

## Multiple video windows

### Multiple video windows

[Prev](#)

[Next](#)

Multiple video windows ([dual display](#) feature "extended" to more than 2 windows).

#### Description

The [Display\\_...](#) and [DualDisplay\\_...](#) features allow to control 2 video windows for the same source (e.g. a preview control embedded in the application, and a 2nd full screen window on another monitor)

It is possible to use more than 2 video windows by invoking directly the [SetDisplay...](#) functions, and by passing the index of the window, in the 2..8 range (0 and 1 are reserved respectively to the [Display...](#) and [DualDisplay...](#) properties)

Use the following sample code for 1 embedded preview (enabled by default, corresponding to the "Display... properties") and additional 2 full screen monitors:

```
Videograbber.SetDisplayActive (2, true);  
Videograbber.SetDisplayActive (3, true);
```

#### Videograbber.StartPreview();

```
Videograbber.SetDisplayMonitor (2, 1);  
Videograbber.SetDisplayMonitor (3, 2);  
Videograbber.SetDisplayFullScreen (2, true);  
Videograbber.SetDisplayFullScreen (3, true);
```

The functions related to the multiple windows are:

[SetDisplayActive](#)

[SetDisplayAlphaBlendEnabled](#)  
[SetDisplayAlphaBlendValue](#)  
[SetDisplayAssociatedRenderer](#)  
[SetDisplayAutoSize](#)  
[SetDisplayAspectRatio](#)

[SetDisplayEmbedded](#)  
[SetDisplayFullScreen](#)  
[SetDisplayMonitor](#)  
[SetDisplayMouseMovesWindow](#)  
[SetDisplayPanScanRatio](#)  
[SetDisplayParent](#)  
[SetDisplayStayOnTop](#)  
[SetDisplayTransparentColorEnabled](#)  
[SetDisplayTransparentColorValue](#)  
[SetDisplayVideoPortEnabled](#)  
[SetDisplayVisible](#)

[SetDisplayLeft](#)  
[SetDisplayTop](#)  
[SetDisplayWidth](#)  
[SetDisplayHeight](#)

[SetDisplayLocation](#)

---

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

## Transparency - Color keying

### Transparency - Color keying

[Prev](#)

[Next](#)

Window transparency overview.

#### Description

#### Alphablending and color keying when the video window is detached from the control

When the video window is detached from the control ([Display\\_Embedded](#) = false), it is possible to superpose another video window and to apply semi transparency AND/OR color keying.

The semi transparency is enabled by [Display\\_AlphaBlendEnabled](#), and the semi transparent value is specified by [Display\\_AlphaBlendValue](#) (in the 0..255 range)

The color keying is enabled by [Display\\_TransparentColorEnabled](#), and the RGB color key value is specified by [Display\\_TransparentColorValue](#)

#### Transparency of another video window when the video window is embedded in the control

To use window transparency, simply enable the [ColorKeyEnabled](#) property, then invoke [SetWindowTransparency](#) to set the color key of your form (or window) to make it transparent when moved over the video window.

You have to pass to [SetWindowTransparency](#) the default TVideoGrabber ColorKey. (you can customize the ColorKey value if needed)

Optionally the color key can be defined by the [ColorKey](#) property.

*Look at the transparency checkbox code in the "display" tab of the MainDemo project for sample code.*

#### See Also

[TOnColorKeyChange](#) [ColorKeyEnabled](#) [OnColorKeyChange](#)

---

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

---

## madVR video renderer

### madVR video renderer

[Prev](#)
[Next](#)

madVR Video Renderer

#### Description

The madVR video renderer is supported by installing the package in a folder of your choice and setting VideoGrabber.VideoRenderer = vr\_madVR.

The filter must be first installed by downloading the madVR package from <http://madvr.com>

After unzipping the package, is it possible to:

- either register the madVR renderer filter in DirectShow by running the install.bat
- either just specify the madVR folder to the [ExtraDLLPath](#) property. In this case no registration of the madVR renderer binaries is needed.

To configure the madVR renderer:

- connect the [OnThirdPartyFilterConnected](#) event, that passes the instance of the filter through the Intf parameter
- invoke Intf.QueryInterface (IMadVRSettings, MadVRSettings)
- invoke the MadVRSetting configuration functions

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## Third-party external renderers

### Third-party external renderers

[Prev](#)
[Next](#)

Third party external video renderers

#### Description

#### Other third-party video renderers

*(if you are using a BlackMagic Decklink, see the [Blackmagic Decklink cards](#) chapter)*

If your video card is shipped with a third-party custom renderer filter, it is possible to let TVideoGrabber use it instead of the default video renderers.

You must pass to TVideoGrabber the CLSID of the filter. You can retrieve it by searching the registry for the name of the filter.

Feel free to contact us at [support@datastead.com](mailto:support@datastead.com) if you have problems to find the CLSID.

Here is some sample code to use a custom renderer filter:

```
...
uses ComObj;
...
procedure TfrmMainForm.Button1Click(Sender: TObject);
const
  CLSID_MyRendererFilter: TGUID = '70e102b0-5556-11ce-97c0-00aa0055595a'; // replace the
begin
  if VideoGrabber.ThirdPartyFilter_AddToList (tpf_VideoRenderer, GUIDToString (CLSID_MyF
    ShowMessage ('filter added');
    VideoGrabber.StartPreview;
  end;
end;
```



## Screen recording - desktop capture

### screen capture and recording with or without cursor

#### screen capture and recording with or without cursor

[Prev](#)[Next](#)

Screen recording

#### Description

#### Screen recording

The screen recording is enabled when [VideoSource](#) = `vs_ScreenRecording`.

Then, all the preview and recording features of the video capture devices can be applied:

- the preview by invoking [StartPreview](#),
- the frame capture by invoking [CaptureFrameTo](#),
- the recording by invoking [StartRecording](#),
- the compression on the fly or after capture (see the **Recording** section)
- the [video processings](#) and [graphics and text overlays](#) when the frame grabber is enabled.

The recording of the cursor is enabled or disabled with [ScreenRecordingWithCursor](#).

The capture of layered windows (e.g. floating or transparent windows) can be enabled by the [ScreenRecordingLayeredWindows](#) property.

If some areas of the screen are not properly recorded, try to enable or disable the [ScreenRecordingThroughClipboard](#) property.

#### WMV screen recording compressed with the Windows Media screen codec

when recording in WMV, the following settings will automatically use the Microsoft's Windows Media screen codec:/line

```
VideoGrabber.VideoSource = vs_ScreenRecording
VideoGrabber.RecordingMethod = rm_ASF
VideoGrabber.ASFProfileVersion = apv_ProfileVersion_9
VideoGrabber.ASFVideoQuality = 95
VideoGrabber.StartRecording()
```

or

```
VideoGrabber.VideoSource = vs_ScreenRecording
VideoGrabber.RecordingMethod = rm_ASF
VideoGrabber.ASFProfileVersion = apv_ProfileVersion_9
VideoGrabber.ASFVideoBitRate = 2000000
VideoGrabber.StartRecording()
```

(note: ASFVideoBitRate and ASFVideoQuality are exclusive, ASFVideoQuality is mandatory)

*If you are using a .prx profile, select the screen codec in the codec list when creating it with WMPProEdt.exe*

#### AVI screen recording compressed by using third-party codecs



The recording will give better results (quality/size) if codecs specifically designed for this purpose are used for the recording.

E.g. by using the Innoheim ISCC codec:/line

```
VideoGrabber.VideoSource = vs_ScreenRecording
VideoGrabber.RecordingMethod = rm_AVI
VideoGrabber.VideoCompressor = VideoGrabber.VideoCompressorIndex ("inno Screen Capture Co
VideoGrabber.CompressionType = ct_Video
if VideoGrabber.VideoCompressor > -1 then
    VideoGrabber.StartRecording()
else
    MessageBox ("screen codec not installed")
end
```

### Specifying the monitor(s) to record when more than 1 monitor is available

If more than one monitor is available, the number of the monitor to record is specified by the [ScreenRecordingMonitor](#) property.

- to record the default monitor set ScreenRecordingMonitor = 0 (default value).
- to record the 2nd monitor, set ScreenRecordingMonitor = 1, and so on...

### Recording the extended desktop

Simply set [ScreenRecordingMonitor](#) = -1, this will record the whole extended desktop across all the monitors.

#### Remarks:

- the preview or recording frame rate of the screen display is specified by the [FrameRate](#) property.
- when recording the full screen, it is recommended to use a low frame rate to prevent dropped frames.
- it is possible to record only a part of the screen by specifying a [cropping area](#).
- when [ScreenRecordingThroughClipboard](#) is enabled, the clipboard is continuously flushed during preview and recording.

#### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

---

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

---

## screen capture and recording of a specified window

### screen capture and recording of a specified window

[Prev](#)

[Next](#)

Screen recording of a window

#### Description

#### Screen recording of a window

It is possible to record only a window, instead of the whole screen.

There are 2 constraints:

- the window may not be resized after the recording starts,
- the window must not be overlapped by another window.

#### Identifying the video window to record

It is necessary to specify to the component the window to be recorded.  
This can be achieved by 2 ways.

#### 1. simple way

- identify the name of the window to record by looking at its title bar, and invoke [SetWindowRecordingByName](#) by passing the whole title bar, or only a substring that identifies it.
- then you can pass:
  - . either the full string to the WindowName parameter with ExactMatch = true
  - . either a substring picked up in the window name, with ExactMatch = false (in this case the string is not case-sensitive).

#### 2. advanced way

Use [EnumerateWindows](#) to retrieve the window name.

After invoking [EnumerateWindows](#), the [OnEnumerateWindows](#) will occur for each visible window, and will return:

- the window name
- the name of the class of the window
- the window handle

Then, after choosing the window, you can specify it by invoking [SetWindowRecordingByHandle](#) and passing the window handle returned by the [OnEnumerateWindows](#) event.

### Starting the window recording

- specify the window to record by using the 1. or 2. method above,
- set [VideoSource](#) = vs\_ScreenRecording
- invoke [StartRecording](#) to record, or [StartPreview](#) and then [CaptureFrameTo](#) to make single window captures.

### Window is destroyed during preview or recording

If the window is destroyed during preview or recording, the preview or recording stops.

### Screen capture and recording of non-visible (hidden) windows

It is possible to enable the capture of non-visible windows by enabling the [ScreenRecordingNonVisibleWindows](#) property.

*Note: this mode may give variable results depending on the content of the window*

---

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

---

## Recording only a part of the screen

### Recording only a part of the screen

[Prev](#)

[Next](#)

Recording only a part of the screen

#### Description

#### Recording only a part of the screen

It is possible to limit the recording to a part of the screen by specifying the width and height that must be recorded at the x, y location.

To activate this feature simply invoke `ScreenRecordingUsingCoordinates` before invoking `StartPreview` or `StartRecording`.

E.g.

```
VideoGrabber.VideoSource := vs_SreenRecording;
VideoGrabber.ScreenRecordingUsingCoordinates (true, 10, 10, 300, 100);
VideoGrabber.StartPreview;
```

Remarks:

- the scLeft and scTop location may be modified dynamically while previewing or recording
- this feature is inactive if the recording of a window has been activated with [SetWindowRecordingByName](#) or [SetWindowRecordingByHandle](#) .

---

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

---

## Desktop capture

### Desktop capture

[Prev](#)

[Next](#)

Desktop capture

#### Description

#### How to perform a desktop capture

- set [VideoSource](#) = vs\_ScreenRecording
- invoke [StartPreview](#)
- minimizes the app (or make it non-visible)
- then it is possible to capture the desktop image to a memory bitmap or a BMP or JPEG file by invoking [CaptureFrameTo](#), or perform a screen recording as explained in the [recording the screen with or without cursor](#) chapter.

This can be quick tested in 2 clicks by using MainDemo.exe included in the package as follows:

- run MainDemo.exe
- in the "video source" tab, select video source = "screen recording"
- click StartPreview
- go to the "frame grabber" tab
- move the MainDemo.exe window at the bottom of the desktop (to hide most of the MainDemo.exe app), just keep the buttons of "one shot frame capture" group visible,
- then click "BMP file" or "JPEG file" to capture the desktop image to a file, or "TBitmap" to capture it to a new window.

Of course, the app must be kept minimized (or not visible) to capture the desktop without capturing your app's window.

Remark: if some special windows appear black in the captured image, enable the [ScreenRecordingThroughClipboard](#) property.

---

Created with the Standard Edition of HelpNDoc: [Transform Your Word Doc into a Professional-Quality eBook with HelpNDoc](#)

---

## Recording on the fly of video clips and live streams

---

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

---

### video clips and live streams recorded on the fly

#### video clips and live streams recorded on the fly

[Prev](#)

[Next](#)

Recording "on the fly" of video clips and live streams

#### Description

#### Preview and recording "on the fly" of video clips and live streams

It is possible to record "on the fly" video clips and live streams.

- 1) first of all set [VideoSource](#) = [vs\\_VideoFileOrURL](#) to select a video clip or a live stream as video source.
- 2) then set the [VideoSource\\_FileOrUrl](#) property with the path or the URL of the video clip or the live stream. Then, all the preview and recording features of the video capture devices can be applied:

#### Default settings

- the audio rendering is enabled by [AudioDeviceRendering](#),
- the frame capture can be used by invoking [CaptureFrameTo](#) if the [frame grabber](#) is enabled.
- the [video processings](#) and [graphic overlays](#) can be applied if the [frame grabber](#) is enabled.

#### Preview

The preview is started by invoking [StartPreview](#).

#### Recording

- the (re)compression can be applied on the fly through the [Datastead Encoder](#)
- the recording is started by invoking [StartRecording](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

---

## Video clips built on the fly from Bitmap handles, BMP or JPEG files

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

---

### Video clips built on the fly by passing bitmap handles, BMP or JPEG files

#### Video clips built on the fly by passing bitmap handles, BMP or JPEG files

[Prev](#)
[Next](#)

How to create a video clip directly from bitmap handles.

#### Description

A video clip can be created on the fly by directly passing either directly memory bitmap handles, either the file name of BMP or JPEG files.

This feature is activated when [VideoSource](#) is set to [vs\\_JPEGsOrBitmaps](#).

The frames are passed by invoking [SendImageToVideoFromBitmaps](#) and passing as parameter:

- either the path to one or the BMP or JPEG files (**ImageFilePath** parameter).
- either a bitmap handle (**BitmapHandle** parameter, see remark (b) below),

**1. BUILDING A VIDEO CLIP BY PASSING IMAGES WHEN THEY ARE RECEIVED, IN REAL TIME** (e.g. the images are currently received from another live source at 15 fps)

- set [VideoSource](#) = [vs\\_JPEGsOrBitmaps](#)
- set [FrameRate](#) with the real frame rate
- invoke [SendImageToVideoFromBitmaps](#) a first time so the component can learn the video format (*if this step is omitted the next step will fail*)
- invoke [StartPreview\(\)](#) or [StartRecording\(\)](#)
- then invoke periodically [SendImageToVideoFromBitmaps](#) or [SendImageToVideoFromBitmaps2](#) as soon as new frames are available to pass to the component.

I

**2. BUILDING A VIDEO CLIP FROM AN EXISTING SET OF IMAGES** (e.g. a set of Jpeg files located in a folder)

When [StartPreview](#) or [StartRecording](#) is invoked, the [OnVideoFromBitmapsNextFrameNeeded](#) occurs periodically (depending of the [FrameRate](#) property), requesting for the next image that will be used as a

video frame to built the video stream.

From this event, invoke [SendImageToVideoFromBitmaps](#) and pass the bitmap handle or the file name of the next frame.

- set [VideoSource](#) = vs\_JPEGsOrBitmaps,
- set [FrameRate](#) with the desired frame rate
- create the [OnVideoFromBitmapsNextFrameNeeded](#) event, and put the code required to pass the bitmap handles, or the the image files,
- from the [OnVideoFromBitmapsNextFrameNeeded](#) event, invoke [SendImageToVideoFromBitmaps](#) or [SendImageToVideoFromBitmaps2](#) and pass as parameter either the file name, either the bitmap handle
- invoke [StartPreview\(\)](#) or [StartRecording\(\)](#)

*You can find sample code in the MainDemo project included in the package.*

#### **Remarks about SendImageToVideoFromBitmaps:**

**a) BitmapHandle and ImageFilePath** are mutually exclusive:

- if you pass a file path to **ImageFilePath**, pass 0 to the **BitmapHandle** paramter
- if you pass a bitmap handle to the **BitmapHandle** parameter, pass an empty string to do not pass a file path to the **ImageFilePath** parameter

**b) if the you pass a BitmapHandle**, enable **CanFreeBitmapHandle** if TVideoGrabber must free the bitmap when it is no longer needed, or disable **CanFreeBitmapHandle** if you need to reuse the bitmap later.

**c) all the images** passed to this event must have the same format.

#### **Remark about the frame rate**

When recording without audio to an AVI file, it is possible to record as fast as possible by setting a very hight frame rate, e.g. [FrameRate](#) = 200) and then set the final frame rate when the recording ends from the [OnEndOfAVIRecording](#) event.

- \* you can apply the [on the fly compression](#) before invoking StartRecording, if needed.
- \* the **FirstSample** parameter is true for the first frame, and false for the others. You can test it e.g. to restart from your first bitmap.
- \* set **EndOfData** to true to notify the end of stream. This is equivalent to invoking [StopRecording](#) or [StopPreview](#).

**c.** if you pass a bitmap handle, no not release it. If the bitmap handle is picked up from a TBitmap object (by passing TBitmap.Handle), be sure to release the handle before freeing the TBitmap.  
E.g.:

```
procedure TForm1.VideoGrabberVideoFromBitmapsNextFrameNeeded(Sender: TObject; FirstSample: Boolean;
var
    Bitmap: TBitmap;
begin
    Bitmap := TBitmap.Create;
    Bitmap.Assign (Image1.Picture);
    BitmapHandle := Bitmap.Handle;
    VideoGrabber. SendImageToVideoFromBitmaps ("", LongInt (BitmapHandle), true, false);
    Bitmap.ReleaseHandle;
    Bitmap.Free;
end;
```

**See Also**

## Video clips from a fixed set of BMP files or JPEG files merged into a clip

### Video clip built from a fixed set of BMP files or JPEG files

#### Video clip built from a fixed set of BMP files or JPEG files

[Prev](#)[Next](#)

How to create a video clip from a set of bitmap or BMP files or JPEG files.

##### Description

**How to create a video clip from a set of bitmap or BMP files or JPEG files.**

First of all, a temporary file must be created from the set of images (step A).

Then, it is possible:

- to preview the video built from temporary file (step B)
- to record a video clip built from the temporary file to an AVI or ASF file (step C)

#### **A. Create a temporary file from the BMP files or JPG files.**

##### **a1. create a set of images as BMP or JPEG files and put them in an empty directory.**

These files must be all of the same format (all BMP or all JPG, same frame width and height, same color resolution).

The format used is the format of the first video frame found in the directory. If other video frames have not the same format, they will be ignored.

(e.g. set StoragePath with a new directory name, go to the "frame capture tab" and capture 20 frames to JPG files in burst mode).

##### **a2. specify the location of the image files:**

assign the directory containing the image files to "[VideoFromImages\\_SourceDirectory](#)". E.g.:

VideoGrabber.VideoFromImages\_SourceDirectory = "c:/MyImagesFolder"

##### **a3. specify a sorting method for the image files (optional)**

choose a [VideoFromImages\\_BitmapSortedBy](#) value. E.g.:

VideoGrabber.VideoFromImages\_BitmapSortedBy = fs\_NameAsc

Note: numeric sequences must be "zero padded". E.g.:

001.bmp

002.bmp

003.bmp

...

009.bmp

010.bmp

011.bmp

...

098.bmp

099.bmp

100.bmp

101.bmp

...

**a4. choose and specify the temporary file that will be created by using the image files**

assign this file name to the [VideoFromImages\\_TemporaryFile](#) property. E.g.  
 VideoGrabber.VideoFromImages\_TemporaryFile = "MyTempFile.dat"

**a5. create the temporary file from the set of bitmaps:**

Invoke [VideoFromImages\\_CreateSetOfBitmaps](#) .

This function will return true upon success and , that you can use immediately in the next steps below, or reuse later.

*Sample code:*

```
procedure TfrmMainForm.CreateTempFileClick(Sender: TObject);
begin
    VideoGrabber.VideoFromImages_BitmapSortedBy := fs_NameAsc;
    VideoGrabber.VideoFromImages_SourceDirectory := 'MyImagesFolder';
    VideoGrabber.VideoFromImages_TemporaryFile := 'MyTempFile.dat';
    VideoGrabber.VideoFromImages_CreateSetOfBitmaps;
end;
```

**B. Preview the video built from temporary file**

**b1. Select the temporary file as live video source**

Set [VideoSource](#) = vs\_VideoFromImages

**b2. specify the temporary file to use**

e.g.: [VideoFromImages\\_TemporaryFile](#) = "MyTempFile.dat"

**b3. specify if the preview must stop at the end of the image files or repeat from the beginning**

e.g. [VideoFromImages\\_RepeatIndefinitely](#) = False

**b4. Choose a frame rate**

E.g. set [FrameRate](#) = 15

**b5. Start the preview**

Invoke [StartPreview](#)

*Sample code:*

```
procedure TfrmMainForm.startpreviewClick(Sender: TObject);
begin
    VideoGrabber.VideoSource := vs_VideoFromImages;
    VideoGrabber.VideoFromImages_TemporaryFile := 'MyTempFile.dat';
    VideoGrabber.VideoFromImages_RepeatIndefinitely := False;
    VideoGrabber.FrameRate := 10;
    VideoGrabber.StartPreview;
end;
```

**C. Record a video clip built from the temporary file to an AVI or ASF file (step C)**

**c1. Select the temporary file as live video source**

Set [VideoSource](#) = vs\_VideoFromImages

**c2. specify the temporary file to use**

e.g.: [VideoFromImages\\_TemporaryFile](#) = "MyTempFile.dat"

**c3. specify if the preview must stop at the end of the image files or repeat from the beginning**

e.g. [VideoFromImages\\_RepeatIndefinitely](#) = False

**c4. Choose a frame rate**

E.g. set [FrameRate](#) = 15

c5. Choose the [recording method](#). Specify if you wish to use a [video and/or audio compressor](#)

## c5. Start the recording

Invoke [StartRecording](#)

*Sample code:*

```
procedure TfrmMainForm.startrecordingClick(Sender: TObject);
begin
    VideoGrabber.VideoSource := vs_VideoFromImages;
    VideoGrabber.VideoFromImages_TemporaryFile := 'MyTempFile.dat';
    VideoGrabber.VideoFromImages_RepeatIndefinitely := False;
    VideoGrabber.StoragePath := 'VideoStorageFolder';
    VideoGrabber.CompressionMode := cm_NoCompression;
    VideoGrabber.FrameRate := 10;
    VideoGrabber.StartRecording;
end;
```

## See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps](#) [SendImageToVideoFromBitmaps2](#)  
[VideoFromImages](#) [BitmapsSortedBy](#) [VideoFromImages\\_RepeatIndefinitely](#)  
[VideoFromImages\\_SourceDirectory](#) [VideoFromImages\\_TemporaryFile](#)

---

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

---

## Video processing

---

Created with the Standard Edition of HelpNDoc: [News and information about help authoring tools and software](#)

---

### Zoom

<b>Zoom</b>	<a href="#">Prev</a>
-------------	----------------------

Zoom

#### Description

The zoom can be activated by specifying a [ZoomCoeff](#) value greater than 1000 (expressed as per thousand).

E.g.:

1000 means no zoom (default value)

2500 means a zoom of 2.5x

4000 means a zoom of 4.x

It is possible to move the X and Y center positions by specifying a [ZoomXCenter](#) and [ZoomYCenter](#) values greater or lower than 0 (expressed in pixels).

**Note:** the [frame grabber](#) must be enabled to use this feature.

## See Also

---

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

---

### Rotation, mirroring, vertical and horizontal flip

<b>Rotation, mirroring, vertical and horizontal flip</b>	<a href="#">Prev</a>	<a href="#">Next</a>
--	----------------------	----------------------

Rotation ,mirror, vertical and horizontal flip



## Description

### Rotation ,mirror, vertical and horizontal flip

The [VideoProcessing\\_Rotation](#) and [VideoProcessing\\_RotationCustomAngle](#) properties offer any combination of rotation and mirror effects.

### Flip

- the [VideoProcessing\\_FlipVertical](#) property flips the video vertically
- the [VideoProcessing\\_FlipHorizontal](#) property flips the video horizontally

### Preventing the drawing to be rotated

When the rotation is used and [OverlayAfterTransform](#) is enabled (e.g. to rotate the text overlay with the video), any drawing performed from the [OnFrameOverlayUsingDC](#) event will be rotated as well.

To retrieve the initial coordinates and prevent the drawing to be rotated, use [RetrieveInitialXYAfterRotation](#).

### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

---

## Cropping and zooming

### Cropping and zooming

[Prev](#)
[Next](#)

Cropping and zooming

### Description

### Cropping

It is possible to easily define an area of the video source that will be cropped.

- define a cropping size with [Cropping\\_Width](#) and [Cropping\\_Height](#),
- enable the cropping with [Cropping\\_Enabled](#),
- start the preview, recording or playback,
- now you can move the cropping area (even dynamically during preview, recording or playback) with [Cropping\\_X](#) and [Cropping\\_Y](#).

Trackbar bounds

When trackbars are used to move the cropped area over the video window, the maximum position for these trackbars are returned by the [Cropping\\_XMax](#) and [Cropping\\_YMax](#) properties.

The best location to update the trackbar bounds is from the [OnResizeVideo](#) event. E.g.:

```
procedure TForm1.VideoGrabberResizeVideo(Sender: TObject; SourceWidth, SourceHeight: Integer)
begin
    TrackBarX.Min := 0;
    TrackBarX.Position := VideoGrabber.Cropping_X;
    TrackBarX.Max := VideoGrabber.Cropping_XMax;

    TrackBarY.Min := 0;
    TrackBarY.Position := VideoGrabber.Cropping_Y;
```

```
    TrackBarY.Max := VideoGrabber.Cropping_YMax;
end;
```

## Zooming

It is possible to zoom a cropped area as follows:

- define a cropping size with [Cropping\\_Width](#) and [Cropping\\_Height](#),
- enable the cropping with [Cropping\\_Enabled](#),
- start the preview, recording or playback,
- now it is possible to specify dynamically a zooming coefficient with [Cropping\\_Zoom](#),
- it is also possible to move the cropping area with [Cropping\\_X](#) and [Cropping\\_Y](#).

Note: if the zooming area is near an edge of the video frame, when changing [Cropping\\_Zoom](#), the video image will "move" to the opposite direction.

To prevent this problem, enable the [Cropping\\_Outbounds](#) property. This will keep the zooming location fixed against the edges, however in counterpart a black border will appear when [Cropping\\_X](#) and [Cropping\\_Y](#) are near the edges.

## See Also

[Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

Created with the Standard Edition of HelpNDoc: [Easily Add Encryption and Password Protection to Your PDFs](#)

## Image adjustments (brightness, contrast, etc)

### Image adjustments (brightness, contrast, etc)

[Prev](#)

[Next](#)

VMR9 image adjustments

#### Description

#### Video stream image adjustments

When the [frame grabber](#) is enabled, it is possible to adjust the brightness, contrast, hue and saturation of the video stream by using:

[VideoProcessing\\_Brightness](#)

[VideoProcessing\\_Contrast](#)

[VideoProcessing\\_Hue](#)

[VideoProcessing\\_Saturation](#)

These properties affect the whole video stream, therefore the captured frames and the recorded streams will be affected.

On the other hand, each of these image processings requires CPU when the corresponding property is "in use" (when its value is  $\neq 0$ ).

#### Greyscale and inverted colors

When the [frame grabber](#) is enabled, it is possible:

- to convert the video stream in greyscale mode by enabling [VideoProcessing\\_GrayScale](#)
- to invert the colors by enabling [VideoProcessing\\_InvertColors](#)

#### Pixellization

The video frames will be pixellized by assigning to [VideoProcessing\\_Pixellization](#) a value greater than 1.

## VMR9 display image adjustments

Unlike the video stream image adjustments above that require CPU, it is possible to adjust the brightness, contrast, hue and saturation at the graphic card level by using the VMR9 video renderer. Of course, this processing affects only the display, and not the captured frames or the recorded streams.

The corresponding functions are:

[GetVMR9ImageAdjustmentBounds](#) -> returns the bounds of a setting (e.g. the brightness), useful e.g. to setup a trackbar

[IsVMR9ImageAdjustmentAvailable](#) -> returns true if a setting (e.g. the brightness) is available

[SetVMR9ImageAdjustmentValue](#) -> sets a setting value (e.g. the brightness).

### See Also

[GetVMR9ImageAdjustmentBounds](#) [IsVMR9ImageAdjustmentAvailable](#) [SetVMR9ImageAdjustmentValue](#)  
[TVMR9ImageAdjustment](#)

---

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

---

## Deinterlacing

### Deinterlacing

[Prev](#)
[Next](#)

#### Deinterlacing

#### Description

#### DEINTERLACING

### A) Deinterlacing through the VMR9 video renderer

If you don't need to capture frames or perform graphics or text overlays, set [FrameGrabber](#) = fg\_Disabled and [VideoRenderer](#) = vr\_VMR9, then the VMR9 will deinterlace the video automatically, to need to enable anything else.

### B) Deinterlacing by enabling the VideoProcessing\_Deinterlacing property

The deinterlacing is activated by the [VideoProcessing\\_Deinterlacing](#) property, that accepts the following values:

**di\_Disabled** : deinterlacing is disabled (embedded)

**di\_HalfSize** : built-in half-size deinterlacing (embedded)

**di\_FullSize** : built-in full size deinterlacing (embedded)

**di\_DScaler** : deinterlacing through the DScaler deinterlacer (third-party filter).

**di\_AVISynth** : deinterlacing through AVISynth (third-party library, AVISynth must be installed)

**di\_FFDSHOW** : deinterlacing through FFDSHOW (third-party library, FFDSHOW must be installed)

**di\_ThirdPartyDeinterlacer**: deinterlacing by using a third-party deinterlacer specified by name to the [ThirdPartyDeinterlacer](#) property

### Deinterlacing through the built-in deinterlacer

**di\_HalfSize**: performs a half-size deinterlacing,

**di\_FullSize**: performs a full size deinterlacing (the [frame grabber](#) must be enabled)

### Deinterlacing through FFDSHOW

- FFDSHOW must be installed (see [Video and Audio decoders](#))

- set [VideoProcessing\\_Deinterlacing](#) = di\_FFDSHOW

then start the preview, recording or playback.

### Deinterlacing through AVISynth

The deinterlacing through AVISynth produces a very good quality, however it is available only for video clips, not for a video capture device being captured.

- download AVISynth 2.5 or higher (the .exe version) from <http://avisynth.org>
- install AVISynth with its default options
- set [VideoProcessing\\_Deinterlacing](#) = di\_AVISynth

then open the clip a normal way, e.g.:

```
VideoGrabber.PlayerFileName = "...the video clip.avi"
VideoGrabber.OpenPlayer
```

and the video clip will be deinterlaced automatically in the background through this library.

### Deinterlacing through a specified deinterlacer filter

Register the deinterlacer filter with regsvr32.exe, then it will appear in the [DirectShowFilters](#) list after restarting the application.

Then invoke:

```
VideoGrabber.ThirdPartyDeinterlacer = "....." // specify the deinterlace filter name as it appears in the
DirectShowFilters list
VideoGrabber.VideoProcessing_Deinterlacing = di_ThirdPartyDeinterlacer
```

### Deinterlacing through DScaler

TVideoGrabber supports the DScaler deinterlacer, however we do not recommend it.

You have to use the **Deinterlace120ax.zip** package that can be downloaded here:  
[http://sourceforge.net/project/showfiles.php?group\\_id=7420](http://sourceforge.net/project/showfiles.php?group_id=7420)

Simply unzip the package and register the filter by running **regsvr32.exe Deinterlace.ax**

The DScaler deinterlacer must be activated as follows:

- set [VideoProcessing\\_Deinterlacing](#) = di\_DScaler
- invoke [ShowDialog \(dlg\\_DScaler\)](#) to adjust the deinterlacer settings.
- then invoke [StartPreview](#) to see to changes.

**Remark: when using a DV source, if you get either:**

- a interlacing that looks like colored snow in 8 bits colors
- frozen video frames (e.g. one each 2 seconds)

**use the following settings:**

- set DVReduceFrameRate = true
- you may have to check or uncheck the "Is Odd field first" checkbox
- you may have to choose another deinterlacing method than the default one.

### **Important notes:**

1. with DV sources, be sure to always select the "full" video size in the [VideoSizes](#) list, whatever the deinterlacing method used.
2. if applying deinterlacing on the recording, be sure to disable it during playback, otherwise it will be applied 2 times, e.g.:
  - > you start DV preview in full size,
  - > you apply half-size deinterlacing during capture, OK.
  - > therefore the AVI clip contains half size video, OK.
  - > you forget to disable the half-size deinterlacing before opening the clip, the half-size deinterlacing is applied a 2nd time -> WRONG

-> the AVI clip appears as quarter-size

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## Motion detection

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

### Motion detection

#### Motion detection

[Prev](#)

[Next](#)

Motion detection.

#### Description

##### Motion detection overview

The motion detection is enabled when [MotionDetector.Enabled](#) is true. The [frame grabber](#) must be enabled to use this feature.

When motion detection is enabled, TVideoGrabber compares the last video frame received to the previous one, unless a reference sample has been specified with [MotionDetector.UseThisReferenceSample](#)

For each video frame, if motion is detected, the [OnMotionDetected](#) event occurs, that returns a global motion ratio, depending of the number of cells in which motion has been detected, and the level of motion in each cell.

If no motion is detected, the [OnMotionNotDetected](#) event occurs.

##### Areas in which motion can be detected

To let you limit the areas of the video frames in which motion will be detected, the video frame is divided in a **grid** composed of lines and rows, that defines cell having a motion level.

In each cell, the motion level can be set in a range from 1 (minimal motion sensitivity) to 9 (maximal motion sensitivity).

A value of 0 disables motion detection in the cell.

This is explained in the [Grid structure / grid sensitivity](#) chapter.

##### Triggered motion detection

By default, the motion detection occurs for each video frame.

If you want to trigger the motion detection manually, enable the [MotionDetector.Triggered](#) property.

Then, invoke [MotionDetector.TriggerNow](#) during preview, recording or playback to trigger the motion detection one time, until the next [MotionDetector.TriggerNow](#) call.

##### Detection of the video signal

Is it possible to detect if the video signal is present or not for each video frame, by invoking

[IsVideoSignalDetected](#).

##### See Also

[Color / Greyscale](#) [Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector.CellMotionRatio](#) [MotionDetector.CompareBlue](#) [MotionDetector.CompareGreen](#) [MotionDetector.CompareRed](#) [MotionDetector.Enabled](#) [MotionDetector.EnumGridDialogControls](#) [MotionDetector.Get2DTextGrid](#) [MotionDetector.Get2DTextMotion](#) [MotionDetector.GetCellLocation](#) [MotionDetector.GetCellSensitivity](#) [MotionDetector.GetCellSize](#) [MotionDetector.GloballyIncOrDecSensitivity](#) [MotionDetector.GlobalMotionRatio](#) [MotionDetector.GreyScale](#) [MotionDetector.Grid](#) [MotionDetector.GridXCount](#) [MotionDetector.GridYCount](#) [MotionDetector.IsGridValid](#) [MotionDetector.MaxDetectionsPerSecond](#) [MotionDetector.ReduceCPULoad](#) [MotionDetector.ReduceVideoNoise](#) [MotionDetector.Reset](#) [MotionDetector.ResetGlobalSensitivity](#) [MotionDetector.SetCellSensitivity](#) [MotionDetector.SetGridSize](#) [MotionDetector.ShowGridDialog](#) [MotionDetector.Triggered](#) [MotionDetector.UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion.Enabled](#)

[RecordingOnMotion](#) [MotionThreshold](#) [RecordingOnMotion](#) [NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

## Motion ratio

### Motion ratio

[Prev](#)

[Next](#)

Motion detection and motion ratio.

#### Description

##### Motion ratio

The motion ratio is the result of motion detection between 2 video frames. All motion ratios values are within the 0..1 range.

For each video frame received:

##### a) if motion is detected:

The [OnMotionDetected](#) event occurs. This event returns the and returns a global motion ratio. From this event you can also query cells using [CellMotionRatio \(x, y\)](#) to get their individual motion ratio.

##### b) if no motion is detected:

The [OnMotionNotDetected](#) event occurs.

#### See Also

[Color / Greyscale](#) [Grid structure / grid sensitivity](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector](#) [CellMotionRatio](#) [MotionDetector](#) [CompareBlue](#) [MotionDetector](#) [CompareGreen](#) [MotionDetector](#) [CompareRed](#) [MotionDetector](#) [Enabled](#) [MotionDetector](#) [EnumGridDialogControls](#) [MotionDetector](#) [Get2DTextGrid](#) [MotionDetector](#) [Get2DTextMotion](#) [MotionDetector](#) [GetCellLocation](#) [MotionDetector](#) [GetCellSensitivity](#) [MotionDetector](#) [GetCellSize](#) [MotionDetector](#) [GloballyIncOrDecSensitivity](#) [MotionDetector](#) [GlobalMotionRatio](#) [MotionDetector](#) [GreyScale](#) [MotionDetector](#) [Grid](#) [MotionDetector](#) [GridXCount](#) [MotionDetector](#) [GridYCount](#) [MotionDetector](#) [IsGridValid](#) [MotionDetector](#) [MaxDetectionsPerSecond](#) [MotionDetector](#) [ReduceCPULoad](#) [MotionDetector](#) [ReduceVideoNoise](#) [MotionDetector](#) [Reset](#) [MotionDetector](#) [ResetGlobalSensitivity](#) [MotionDetector](#) [SetCellSensitivity](#) [MotionDetector](#) [SetGridSize](#) [MotionDetector](#) [ShowGridDialog](#) [MotionDetector](#) [Triggered](#) [MotionDetector](#) [UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion](#) [Enabled](#) [RecordingOnMotion](#) [MotionThreshold](#) [RecordingOnMotion](#) [NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## Color intensity

### Color intensity

[Prev](#)

[Next](#)

Color intensity

#### Description

##### Color intensity of a cell

It is possible to retrieve the average color intensity of a cell by invoking [MotionDetector](#) [CellColorIntensity](#) ([RGBSelector](#), x, y).

The function returns the average intensity for the specified color in the specified cell.

The value returned is in the [0.255] range.

##### Global intensity of a color

The [MotionDetector](#) [GlobalColorIntensity](#) ([RGBSelector](#)) function returns the global average intensity for the specified color, in the [0.255] range.

## Grid structure / grid sensitivity

### Grid structure / grid sensitivity

[Prev](#)[Next](#)

Grid structure.

#### Description

##### Overview

A sensitivity grid is applied on video frames. This grid allows to choose areas within the image where motion must be detected. The grid divides the image into rows and columns, defining cells. Each cell sensitivity can be set from 0 (no motion detection) to 9 (maximal motion sensitivity).

E.g. if the video frame is 320 x 240, and you specify a grid size of 32x24, each cell will have a 10x10 size.

One motion ratio is calculated per cell, and the [OnMotionDetected](#) event reports the X and Y coordinates of the cell that reported the higher motion ratio.

Common grids sizes span 2 x 2 to 30 x 30, but any grid size can be used, from 1 x 1 to the width x height size of the video frame.

The grid is defined by a simple string that can be set and retrieved by using the [Grid](#) property, allowing to easily save it and retrieve it using the registry (see Grid Structure below).

#### Grid setup

The sensitivity of each cell can be set programmatically by assigning the [Grid](#) property, or interactively through a dialog invoked by [ShowGridDialog](#). This dialog shows the grid superposed to the current video image, allowing to enable/disable the motion detection in each cell by clicking on the cell.

#### Grid structure

The grid corresponds to a division of the frame into rows and columns, equally spaced. The minimum size is 1 x 1, the maximum size is the size of the frame.

The grid divides the video frame in cells. A value associated to each cell represents the sensitivity of this cell, from 0 (no motion sensitivity) to 9 (maximal motion sensitivity).

Grids samples:

2 x 2 grid:

```
0 6
6 6
```

10 x 8 grid:

```
0 0 0 0 0 0 0 0 0 0
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 7
```

4 x 8 grid:

```
8 8 0 0
8 8 0 0
8 8 0 0
8 8 0 0
8 8 0 0
8 8 0 0
8 8 0 0
8 8 0 0
```



8 8 0 0

If the grid over sizes the frame (the grid has more columns that the frame width and/or more rows than the frame height, in pixels), the motion detection is done on the top/left part of the grid that covers the frame, the parts that oversize the frame are ignored, and the OnGridOversizesFrame event occurs returning the delta width and/or height in pixels that do not cover the frame.

### Grid layout

The grid is set and retrieved as a linear string layout of each cell sensitivity. This layout allows to easily store the grid in the registry and retrieve it as a simple string.

The grid string begins by the 1st row, a blank separator, then the 2nd row, a blank separator, the 3rd, row, and so on. E.g., for a 10 x 10 grid :

"0000044441 5555555551 4444444441 5555555551 6666666661 6666666661 6666666661 3333333331 0000000000 0000000000"

corresponds to the following 2D grid:

```
0 0 0 0 0 4 4 4 4 1
5 5 5 5 5 5 5 5 5 1
4 4 4 4 4 4 4 4 4 1
5 5 5 5 5 5 5 5 5 1
6 6 6 6 6 6 6 6 6 1
6 6 6 6 6 6 6 6 6 1
6 6 6 6 6 6 6 6 6 1
3 3 3 3 3 3 3 3 3 1
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

In this sample, the 4 top/left cells and the 2 bottom rows will not detect motion, and the most sensitive part of the frame is located on the 5th, 6th and 7th rows.

### See Also

[Color / Greyscale Motion ratio Recording only when motion is detected Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_ShowGridDialog MotionDetector\\_Triggered MotionDetector\\_UseThisReferenceSample OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected RecordingOnMotion\\_Enabled RecordingOnMotion\\_MotionThreshold RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Reach: Convert Your Word Document to an ePub or Kindle eBook

## Color / Greyscale

### Color / Greyscale

[Prev](#)

[Next](#)

Color / greyscale.

#### Description

#### Color sensitivity

The motion detection is performed by default on a RGB basis.



Any of the 3 RGB colors can be enabled or disabled for motion detection by using [CompareBlue](#) , [CompareRed](#) or [CompareGreen](#) .

### Greyscale comparison

To perform motion detection on a greyscale basis, enable the [GreyScale](#) property. In this case CompareBlue, CompareGreen and CompareRed are ignored.

### See Also

[Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion

## Video noise

### Video noise

[Prev](#)
[Next](#)

Video noise.

#### Description

#### Video noise

When the camera captures video frames in a dark environment, it is possible that the electrical background noise of the CCD video cell creates fake motion detection events.

It is possible to reduce the sensitivity of to the video noise by enabling the [ReduceVideoNoise](#) property. Usually this saves about one point of [grid sensitivity](#) (e.g. if motion detection occurred with a grid sensitivity starting from 9, when enabling this property it occurs only with a grid sensitivity starting from 8).

### See Also

[Color / Greyscale](#) [Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Produce Kindle eBooks easily

## Recording only when motion is detected

### Recording only when motion is detected

[Prev](#)
[Next](#)

How to record only when motion is detected.

### Description

To start recording only when motion is detected:

- enable [MotionDetector\\_Enabled](#)
- enable [RecordingOnMotion\\_Enabled](#)
- adjust [RecordingOnMotion\\_MotionThreshold](#) and [RecordingOnMotion\\_NoMotionPauseDelayMs](#) if required,
- disable [RecordingPauseCreatesNewFile](#) to store all the recording in a single AVI file, or enable [RecordingPauseCreatesNewFile](#) to create a new AVI file each time no motion is detected,
- then invoke [StartRecording](#).

[RecordingOnMotion\\_Enabled](#): specifies to record only when motion is detected

[RecordingOnMotion\\_MotionThreshold](#): specifies the minimum motion ratio to (re)start the recording

[RecordingOnMotion\\_NoMotionPauseDelayMs](#): specifies the delay after which the recording is switched back in a paused state when no motion is detected

E.g.:

```
VideoGrabber1->MotionDetector_Enabled = true;
VideoGrabber1->RecordingCanPause = true;
VideoGrabber1->RecordingOnMotion_Enabled = true;
VideoGrabber1->RecordingOnMotion_NoMotionPauseDelayMs = 2000;
VideoGrabber1->RecordingOnMotion_MotionThreshold = 0.001;
VideoGrabber1->RecordingPauseCreatesNewFile := true;
VideoGrabber1->StartRecording();
```

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_ShowGridDialog MotionDetector\\_Triggered MotionDetector\\_UseThisReferenceSample OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected RecordingOnMotion\\_Enabled RecordingOnMotion\\_MotionThreshold RecordingOnMotion\\_NoMotionPauseDelayMs](#)

---

Created with the Standard Edition of HelpNDoc: Make Help Documentation a Breeze with a Help Authoring Tool

---

## Mixing several video sources

---

Created with the Standard Edition of HelpNDoc: Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer

---

### Mixing several video sources into a single one

## Mixing several video sources into a single one

[Prev](#)

[Next](#)

Mixing several video sources into a single one

## Description

### Overview

It is possible to mix one or several TVideoGrabber components **used as "normal" video sources** (video capture devices and/or video clips) into a single TVideoGrabber component used as **mixer**.

This mixer component works independently of the source component, it can be stopped, previewed, recorded, paused, etc... while the 1st component sends it the video frames continuously.

The sources components may be displayed into the mixer component:

- as a basic "copy": a 2nd component in mixer mode receives and display the video from the 1st component
- by switching several sources into a single one when needed
- as a mosaic layout (e.g. 4 cameras displayed at the same time on a 2x2 layout),
- as an alternated display (e.g. the 4 cameras are displayed alternatively one after the other),
- in a mosaic/alternated layout (e.g. the 16 cameras are displayed in 4 alternated mosaic layouts of 2x2 cameras).

### Combined mosaic / alternated mixing

It is possible to combine the alternated and mosaic modes, e.g. to display 16 cameras by groups of 4 cameras displayed alternatively into a 4 X 4 video window.

Parameters of the Mixer\_AddToMixer component

- the 1st parameter is the UniqueId of the source component
- the 2nd parameter is not used for the moment, just set it to 0
- the 3rd parameter is the mosaic line where the source will be displayed (set it to 0 for an "alternated only" use).
- the 4th parameter is the mosaic column where the source will be displayed (set it to 0 for an "alternated only" use).
- the 5th parameter is the display group number (set it to 0 for a "mosaic only" use)
- the 6th parameter is the display group duration in milliseconds (set it to 0 for a "mosaic only" use)
- the 7th parameter should be set to TRUE
- the 8th parameter should be set to TRUE

See [Mixer\\_AddToMixer](#).

### Basic mixing (a 2nd component receives the video frames from the 1st component)

This mode lets you have a 2nd component that uses the first component as video source.  
E.g. the first component makes the preview and the 2nd component starts/stops/pause/resume the recording independently of the 1st component that does a continuous preview.

E.g.:

```
VideoGrabber1.VideoSource = vs_VideoCaptureDevice
VideoGrabber1.StartPreview()
VideoGrabber2.VideoSource = vs_Mixer
VideoGrabber2.Mixer_AddToMixer (VideoGrabber1.UniqueID, 0, 0, 0, 0, 0, true, true);
VideoGrabber2.StartPreview()
```

and VideoGrabber2 will receive as video source the video displayed and sent by VideoGrabber1.

### Choosing the mixer video size

By default, the mixer uses the following video size:

- if the source has been started before starting the mixer, the mixer uses the source size,
- if the source has not been started, the mixer starts in 320x240.

To choose the video size, invoke `UseNearestVideoSize` on the mixer before `StartPreview` or `StartRecording`.

*E.g. to start the mixer in 640x480, invoke:*

```
VideoGrabberMixer.VideoSource = vs_Mixer
VideoGrabberMixer.UseNearestVideoSize (640, 480, true)
VideoGrabberMixer.StartPreview()
```

### Switching several sources into a single one when needed

Similar to the basic mixing with more than one video source, the sources can be switched by invoking [Mixer Activation](#) , e.g:

*1. start the preview of the 1st capture device:*

```
VideoGrabber1.VideoSource = vs_VideoCaptureDevice
VideoGrabber1.VideoDevice = 0
VideoGrabber1.StartPreview()
```

*2. start the preview of the 2nd capture device:*

```
VideoGrabber2.VideoSource = vs_VideoCaptureDevice
VideoGrabber2.VideoDevice = 1
VideoGrabber2.StartPreview()
```

*3. start the 3rd component that will make the preview or recording in mixer mode:*

```
VideoGrabber3.VideoSource = vs_Mixer
int MixerId1 = VideoGrabber3.Mixer_AddToMixer (VideoGrabber1.UniqueID, 0, 0, 0, 0, 0, true)
int MixerId2 = VideoGrabber3.Mixer_AddToMixer (VideoGrabber2.UniqueID, 0, 0, 0, 0, 0, true)
VideoGrabber3.Mixer_Activation (MixerId2, false) // let' start with MixerId1 activated
VideoGrabber3.StartPreview()
```

*4. then to switch between the inputs, activate one and deactivate the other:*

```
VideoGrabber3.Mixer_Activation (MixerId1, false)
VideoGrabber3.Mixer_Activation (MixerId2, true)
or
VideoGrabber3.Mixer_Activation (MixerId2, false)
VideoGrabber3.Mixer_Activation (MixerId1, true)
```

### Activating the mixer component in automatic alternated mixing mode

Let's take an example where the mixer component is named "Mixer1", and the sources components "Source1", "Source2" and "Source3".

- set Mixer1.[VideoSource](#) = vs\_Mixer
- set Mixer1.[Display\\_AutoSize](#) = false (if you want to control the Width and Height of the component and prevent it to be resized automatically)
- set Mixer1.[Mixer\\_MosaicLines](#) = 1
- set Mixer1.[Mixer\\_MosaicColumns](#) = 1
- invoke [Mixer\\_AddToMixer](#) (Source component, 0, 0, 0, **group number**, **group display duration**, true, true).

*E.g if (group 1 = 1500 ms, group 2 = 2000 ms, group 3 = 2500 ms)*

```
Mixer1.Mixer_AddToMixer (Source1.UniqueID, 0, 0, 0, 1, 1500, True, True)
Mixer1.Mixer_AddToMixer (Source2.UniqueID, 0, 0, 0, 2, 2000, True, True)
```

```
Mixer1.Mixer_AddToMixer (Source3.UniqueID, 0, 0, 0, 3, 2500, True, True)
```

then invoke e.g.:

```
Source1.StartPreview()  
Source2.StartPreview()  
Mixer1.StartPreview()
```

See the remark below.

### Activating the mixer component in mosaic mixing mode

In this mode the destination component window is splitted into x lines and y columns, and each source is displayed at a predefined (x,y) location.

Let's take an example where the mixer component is named "Mixer1", and the sources components "Source1", "Source2", "Source3" and "Source4" will be displayed in a 2 x 2 layout.

- set Mixer1.[VideoSource](#) = vs\_Mixer
- set Mixer1.[Display\\_AutoSize](#) = false (if you want to control the Width and Height of the component and prevent it to be resized automatically)
- set Mixer1.[Mixer\\_MosaicLines](#) = 2
- set Mixer1.[Mixer\\_MosaicColumns](#) = 2
- invoke [Mixer\\_AddToMixer](#) (Source component, 0, **line**, **column**, 0, 0, true, true).

```
Mixer1.Mixer_AddToMixer (Source1.UniqueID, 0, 1, 1, 0, 0, True, True)  
Mixer1.Mixer_AddToMixer (Source2.UniqueID, 0, 1, 2, 0, 0, True, True)  
Mixer1.Mixer_AddToMixer (Source3.UniqueID, 0, 2, 1, 0, 0, True, True)  
Mixer1.Mixer_AddToMixer (Source4.UniqueID, 0, 2, 2, 0, 0, True, True)
```

then invoke e.g.:

```
Source1.StartPreview()  
Source2.StartPreview()  
Source3.StartPreview()  
Source4.StartPreview()  
Mixer1.StartPreview()
```

See the remark below.

### Activating the mixer component in alternated/mosaic mixing mode

In this mode each source is displayed alternatively into a single video window.

Let's take an example where the mixer component is named "Mixer1", and the sources components "Source1", "Source2", "Source3" and "Source4" will be displayed alternatively in 2 layouts of 1 x 2 source components.

We will use 2 groups named "55" and "66" (the number is not significant, it must be simply different for each group). The group 55 will be displayed with a duration of 1500 milliseconds, and the group 66 with a duration of 2500 milliseconds.

- set Mixer1.[VideoSource](#) = vs\_Mixer
- set Mixer1.[Display\\_AutoSize](#) = false (if you want to control the Width and Height of the component and prevent it to be resized automatically)
- set Mixer1.[Mixer\\_MosaicLines](#) = 1
- set Mixer1.[Mixer\\_MosaicColumns](#) = 2
- invoke [Mixer\\_AddToMixer](#) (Source component, 0, **line**, **column**, **group number**, **group duration**, true, true).

```
Mixer1.Mixer_AddToMixer (Source1.UniqueID, 0, 1, 1, 55, 1500, True, True)
Mixer1.Mixer_AddToMixer (Source2.UniqueID, 0, 1, 2, 55, 1500, True, True)
Mixer1.Mixer_AddToMixer (Source3.UniqueID, 0, 1, 1, 66, 2500, True, True)
Mixer1.Mixer_AddToMixer (Source4.UniqueID, 0, 1, 2, 66, 2500, True, True)
```

then invoke e.g.:

```
Source1.StartPreview()
Source2.StartPreview()
Source3.StartPreview()
Source4.StartPreview()
Mixer1.StartPreview()
```

**Remark:** you can also invoke [Mixer1.StartPreview](#) before starting or associating the sources, in this case an empty window with a [BackgroundColor](#) will be displayed.

### See Also

[Mixer\\_Activation](#) [Mixer\\_AddToMixer](#) [Mixer\\_MosaicColumns](#) [Mixer\\_MosaicLines](#) [Mixer\\_RemoveFromMixer](#) [Mixer\\_SetOverlayRoundedCorner](#) [Mixer\\_SetupPIPFfromSource](#)

---

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

---

## Reencoding

---

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

### Reencoding of clips in batch mode

#### Reencoding of clips in batch mode

[Prev](#)
[Next](#)

How to reencode video, audio/video or audio clips.

#### Description

#### How to reencode video audio/video or audio clips

TVideoGrabber lets you reencode clips:

- 1) you can simply cut video clips by specifying a start and stop time,
- 2) you can reencode a clip:
  - by specifying a start and or stop time,
  - by using the current audio and/or video compressor,
  - by applying any of the frame grabber features (text overlay, graphics overlay, video rotation, deinterlacing, etc...).

#### Reencoding properties

[Reencoding\\_SourceVideoClip](#): file name of the source video clip

[Reencoding\\_NewVideoClip](#): file name of the video clip to create

[Reencoding\\_StartTime](#): start time expressed in 100ns units, (default -1 = beginning),

[Reencoding\\_StartFrame](#): start frame (default -1 = beginning)

[Reencoding\\_StopTime](#): stop time expressed in 100ns units, (default -1 = end),

[Reencoding\\_StopFrame](#): stop frame (default -1 = end of the clip)

[Reencoding\\_IncludeAudioStream](#): if enabled, the audio stream is included in the new video clip,

[Reencoding\\_IncludeVideoStream](#): if enabled, the video stream is included in the new video clip,

[Reencoding\\_Method](#): rm\_AVI to record in AVI format, or rm\_ASF to record in ASF format,

[Reencoding\\_UseAudioCompressor](#): if enabled, the current [audio compressor](#) is used,

[Reencoding\\_UseVideoCompressor](#): if enabled, the current [video compressor](#) is used,

[Reencoding\\_UseFrameGrabber](#): if enabled, graphics, text overlays, cropping and rotation will be applied,

[Reencoding\\_WMVOutput](#): the clip will be created as .WMV

**About the start/stop frames and times**

- the default -1 value specifies "start from the beginning of the clip" or "stop when the end of the clip is reached".
- the times are specified in 100ns units, e.g. 3 seconds = 30000000
- if you specify a [Reencoding\\_StartTime](#) AND a [Reencoding\\_StartFrame](#), the [Reencoding\\_StartTime](#) will be ignored.
- if you specify a [Reencoding\\_StopTime](#) AND a [Reencoding\\_StopFrame](#), the [Reencoding\\_StopTime](#) will be ignored.

**To start reencoding:**

To start reencoding a video clip:

- set the "Reencoding\_..." properties,
- invoke [StartReencoding](#)

**Important remark:**

when invoking [StartReencoding](#), the process starts the reencoding and **returns immediatly, it does not wait for the reencoding process to complete.**

If you are creating the component programmatically, be sure to wait for the OnReencodingCompleted before destroying the component, otherwise the reencoding process would be interrupted before finished.

E.g.:

```
VideoGrabber1.Reencoding_SourceVideoClip = "MyVideoClipToReencode.avi"
VideoGrabber1.Reencoding_NewVideoClip = "MyReencodedVideoClip.wmv"
VideoGrabber1.Reencoding_WMVOutput = true // output clip is wmv
VideoGrabber1.Reencoding_Method = rm_ASF
VideoGrabber1.Reencoding_StartFrame = -1 // -1 = beginning of the clip. E.g. if you set
VideoGrabber1.Reencoding_StopFrame = -1 // -1 = beginning of the clip. E.g. if you set
VideoGrabber1.Reencoding_StartTime = -1 // -1 = end of the clip. E.g. if you set 50000
VideoGrabber1.Reencoding_StopTime = -1 // -1 = end of the clip. E.g. if you set 100000
VideoGrabber1.Reencoding_IncludeAudioStream = true // if audio stream needed in the re
VideoGrabber1.Reencoding_IncludeVideoStream = true // if video stream needed in the re
VideoGrabber1.Reencoding_UseAudioCompressor = false
VideoGrabber1.Reencoding_UseVideoCompressor = false
VideoGrabber1.Reencoding_UseFrameGrabber = true
VideoGrabber1.StartReencoding // from now the progress will be returned periodically h
```

To stop the reencoding process before it ends, simply invoke [StopReencoding](#).

**Reencoding progress**

When the reencoding begins, the [OnReencodingStarted](#) event occurs.

During the reencoding, the [OnReencodingProgress](#) event occurs periodically, reporting the percentage of completion.

When the reencoding ends, the [OnReencodingCompleted](#) event occurs.

**Reencoding an audio clip / Extracting the audio from a video clip**

It is possible to reencode an audio clip, or to extract only the audio to a .WAV or .MP3 audio clip, just by specifying ".wav" or ".mp3" as file extension for the new clip.

To extract in MP3, first download and register (with regsvr32.exe) the LAME Audio Encoder (freeware) that you can download [here](#) (the direct download link is [here](#))

*Sample code to extract 4 seconds of a WAV audio clip starting at 2 seconds (therefore stop time = 2s + 4s) and save them in MP3:*

```
VideoGrabber.Reencoding_SourceVideoClip = "c:/myfolder/myvideoclip.wav"
```



```
VideoGrabber.Reencoding_Start_Frame = -1
VideoGrabber.Reencoding_Stop_Frame = -1
VideoGrabber.Reencoding_Start_Time = 20000000
VideoGrabber.Reencoding_Stop_Time = 60000000
VideoGrabber.Reencoding_NewVideoClip = "c:/myfolder/mynewclip.mp3"
VideoGrabber.StartReencoding()
```

*Sample code to exact the whole audio from a video clip to a .WAV format:*

```
VideoGrabber.Reencoding_SourceVideoClip = "c:/myfolder/myvideoclip.avi"
VideoGrabber.Reencoding_Start_Frame = -1
VideoGrabber.Reencoding_Stop_Frame = -1
VideoGrabber.Reencoding_Start_Time = -1
VideoGrabber.Reencoding_Stop_Time = -1
VideoGrabber.Reencoding_NewVideoClip = "c:/myfolder/mynewclip.wav"
VideoGrabber.StartReencoding()
```

*Sample code to exact the first 3 seconds of a video clip to a .MP3 format:*

```
VideoGrabber.Reencoding_SourceVideoClip = "c:/myfolder/myvideoclip.avi"
VideoGrabber.Reencoding_Start_Frame = -1
VideoGrabber.Reencoding_Stop_Frame = -1
VideoGrabber.Reencoding_Start_Time = -1
VideoGrabber.Reencoding_Stop_Time = 30000000
VideoGrabber.Reencoding_NewVideoClip = "c:/myfolder/mynewclip.mp3"
VideoGrabber.StartReencoding()
```

Remark:

when extracting audio, only the following Reencoding... settings are taken in account:

[Reencoding\\_StartTime](#)  
[Reencoding\\_StopTime](#)  
[Reencoding\\_StartFrame](#)  
[Reencoding\\_StopFrame](#)

the other Reencoding... settings are ignored.

## Converting a DVD to a M2V file

You can reencode a DVD into a M2V format through DGMPGDec as follows:

- download the DGMPGDec executables from <http://neuron2.net/dgmpgdec/dgmpgdec.html>, unzip in the folder of your choice
- use the following reencoding commands:

*E.g. let's suppose:*

- you have unzipped DGMPGDec into c:/myDGfolder
- your DVD is placed in the "D:" drive
- you want to generate the M2V to c:/myOutputFolder/myOutputVideo.m2v

```
VideoGrabber.Reencoding_SourceVideoClip = "c:/myDGfolder/DGIndex.exe D:/VIDEO_TS"
VideoGrabber.Reencoding_NewVideoClip "c:/myOutputFolder/myOutputVideo.m2v"
VideoGrabber.StartReencoding()
```

**Remark about the reencoding progress:** the values reported by the [OnReencodingProgress](#) event are not significant, because it is not possible to predict the time required to perform the reencoding with DGIndex.exe. The end of the reencoding will be notified by the [OnReencodingCompleted](#) event.

## See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#)  
[Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#)



[Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#)  
[Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#)  
[Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## Reencoding of clips in preview or recording mode

### Reencoding of clips in preview or recording mode

[Prev](#)
[Next](#)

Video clips reencoding in preview/recording mode

#### Description

#### Video clips reencoding in preview/recording mode

The reencoding of video clips can be done as follows:

- set [VideoSource](#) = vs\_VideoFileOrURL,
- set [VideoSource\\_FileOrURL](#) with the file name or the URL of the video clip,
- set all the text overlays, graphics overlays, video effects, etc... you need,
- select the compression settings (see the [Software compression using codecs](#) chapter),
- invoke [StartRecording](#) to record to a new video clip, or [StartPreview](#) to just play it.

Note: a start time and/or stop time can be specified by [VideoSource\\_FileOrURL\\_StartTime](#) and [VideoSource\\_FileOrURL\\_StopTime](#).

#### Reencoding of a playlist in preview/recording mode

You can reencode a playlist as follows:

E.g.:

```
videograbber.Playlist (pl_Clear, '')
videograbber.Playlist (pl_Add, 'vg000004.avi')
videograbber.Playlist (pl_Add, 'vg000002.avi')
videograbber.Playlist (pl_Add, ...
videograbber.Playlist (pl_Add, ...
videograbber.VideoSource = vs_VideoFileOrURL
videograbber.VideoSource_FileOrURL = 'PLAYLIST'
videograbber.StartRecording()
```

You can invoke StartPreview instead of StartRecording to just play the playlist.

#### See Also

[VideoSource\\_FileOrURL\\_VideoSource\\_FileOrURL\\_StartTime\\_VideoSource\\_FileOrURL\\_StopTime](#)

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## Merging or splitting video clips

### Merging or splitting video clips

[Prev](#)
[Next](#)

Merge or split video clips

#### Description

#### To merge clips

Clips can be merged by specifying a playlist, then invoking [StartReencoding](#) through the [PlayList](#) feature.

E.g.:

```
videograbber.Playlist (pl_Clear, '')
videograbber.Playlist (pl_Add, 'vg000004.avi')
videograbber.Playlist (pl_Add, 'vg000002.avi')
videograbber.Reencoding_SourceVideoClip = 'PLAYLIST'
videograbber.Reencoding_NewVideoClip = 'mynewclip.avi'
videograbber.StartReencoding()
```

Depending of the kind of clip reencoded, you can also specify a start/stop time:

```
videograbber.Playlist (pl_Clear, '')
videograbber.Playlist (pl_Add, 'vg000004.avi')
VideoGrabber.PlayList (pl_SpecifyPositions, 'vg000004.avi');
VideoGrabber.OpenPlayerAtTimePositions (50000000, 120000000, true, true);
videograbber.Playlist (pl_Add, 'vg000002.avi')
VideoGrabber.PlayList (pl_SpecifyPositions, 'vg000002.avi');
VideoGrabber.OpenPlayerAtTimePositions (50000000, 120000000, true, true);
videograbber.Reencoding_SourceVideoClip = 'PLAYLIST'
videograbber.Reencoding_NewVideoClip = 'mynewclip.avi'
videograbber.StartReencoding()
```

See the [Reencoding](#) chapter for the reencoding parameters (reencoding method, etc...)

### To split video clips

To split a clip, use the [Reencoding](#) feature by specifying a start time and/or a stop time (or a start frame and/or stop frame).

---

Created with the Standard Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

---

## Video capture devices with multiplexed inputs

---

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

---

### Video capture devices having multiplexed inputs

#### Video capture devices having multiplexed inputs

[Prev](#)
[Next](#)

Video capture cards with multiplexed inputs

#### Description

#### Video capture cards with multiplexed inputs

TVideoGrabber supports video capture boards having multiplexed inputs.

This mode is activated by the [MultiplexedRole](#) property. Each input can be enabled/disabled with [EnableMultiplexedInput](#).

2 cases:

#### a. the video capture card is switching its inputs automatically

If the inputs are switched by the card itself (you see the video frames of each channel alternatively in the same video window), disable the [MultiplexedInputEmulation](#) property.

It is possible that this feature is only available through one of the video inputs. In this case you have to

select this [VideoInput](#) in the [VideoInputs](#) list.

### **b. the video capture card does not switch its inputs automatically**

Whatever the [VideoInput](#) that you select in the [VideoInputs](#) list, you see only one video channel at the time. In this case enable the [MultiplexedInputEmulation](#) property to get the inputs switched programmatically by TVideoGrabber.

Each video input will be switched after a delay specified by the [MultiplexedSwitchDelay](#) (default value: 0 means the input is switched after each video frame received).

When an input is switched, the [MultiplexedStabilizationDelay](#) waits for the specified duration (expressed in milliseconds) to wait for the input stabilization (default value = 70 milliseconds, minimum value = 30 ms).

It is possible to work in 2 ways:

#### **1) one single TVideoGrabber component shows a mosaic layout**

In this mode a single TVideoGrabber component displays all inputs in a single video window as a "mosaic layout".

This mode is activated by setting:

- [MultiplexedRole](#) = mr\_MultiplexedMosaic4 -> for a 4 inputs display
- [MultiplexedRole](#) = mr\_MultiplexedMosaic16 -> for a 16 inputs display

Note: the frame rate is specified by the [FrameRate](#) property.

#### **2) master/slaves: one master TVideoGrabber component distributes the inputs to several TVideoGrabber slaves components**

- in this video capture board is managed in preview mode by the TVideoGrabber master component
- the slaves TVideoGrabber components are "virtual input" components associated to the TVideoGrabber master component.

To activate this mode for a 4 inputs board:

a)

- put a TVideoGrabber component on the form (e.g. VideoGrabber1)
- set its [MultiplexedRole](#) property to mr\_MultiplexedMaster

b)

- put 4 TVideoGrabber components on the form (e.g. VideoGrabber2, VideoGrabber3, VideoGrabber4, VideoGrabber5).
- set their [MultiplexedRole](#) property to mr\_MultiplexedSlave.

c)

- invoke programmatically [AssociateMultiplexedSlave](#) on the master component for each slave component. E.g. add the following code to associate the slaves components to the master component: (in the FormShow event, or before invoking StartPreview):

```
procedure TForm1.FormShow(Sender: TObject);
begin
    VideoGrabber1.AssociateMultiplexedSlave (0, VideoGrabber2.UniqueID);
    VideoGrabber1.AssociateMultiplexedSlave (1, VideoGrabber3.UniqueID);
    VideoGrabber1.AssociateMultiplexedSlave (2, VideoGrabber4.UniqueID);
    VideoGrabber1.AssociateMultiplexedSlave (3, VideoGrabber5.UniqueID);
end;
```

Note: the first parameter is the number of the video input to associate to the slave component.

Then you can invoke [StartPreview](#) to start distributing the virtual inputs to the slaves components.

Notes:

- the frame rate of the master component is specified by the [FrameRate](#) property.
- the frame rate of the slaves components can be adjusted by setting their respective [FrameRate](#) property.

#### **See Also**

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#)

[MultiplexedRole](#) [MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## Opening a clip or an IP URL from a background thread without blocking

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

### Opening a clip or an IP URL from a background thread without blocking the main thread

#### Opening a clip or an IP URL from a background thread without blocking the main thread

[Prev](#)

[Next](#)

Opening a clip or an IP URL from a background thread without blocking the main thread

##### Description

It is possible to open a clip or an URL without blocking the main thread.

##### C# and VB.NET:

- define a class derived from VidGrab.VideoGrabberThread
- put the TVideoGrabber initialization code in the VideoGrabberThreadFunc() class
- synchronize the TVideoGrabber event callbacks that refer to .NET controls with the Invoke method

The "AsynchronousStartFromThread" C# or VB demo project included in the package shows the required sample code to open an IP camera URL.

##### See Also

[Player features](#) [TOnThreadSync](#) [TThreadSyncPoint](#) [EnableThreadMode](#) [OnThreadSync](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## Synchronization of TVideoGrabber components

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

### Synchronization of several TVideoGrabber components

#### Synchronization of several TVideoGrabber components

[Prev](#)

[Next](#)

Synchronization of several TVideoGrabber components.

##### Description

##### SYNCHRONIZATION ACTIVATION

It is possible to synchronize several TVideoGrabber components placed on the same form, or on different forms, within the same application.

This feature is activated by enabling the [Synchronized](#) property on each TVideoGrabber component that must be synchronized with the others.

##### PREVIEW/RECORDING SYNCHRONIZATION

The preview and recording synchronization lets TVideoGrabber prepare the preview or recording on all the specified components, then the 3 components are started by just invoking [StartSynchronized\(\)](#) on one of them.

E.g. to start the preview concurrently on 3 TVideoGrabber components:

```
VideoGrabber1.VideoDevice = VideoGrabber.VideoDeviceIndex ("... name of your 1st video capture device ...")
VideoGrabber2.VideoDevice = VideoGrabber.VideoDeviceIndex ("... name of your 2nd video capture device ...")
VideoGrabber3.VideoDevice = VideoGrabber.VideoDeviceIndex ("... name of your 3rd video capture device ...")
```

```
VideoGrabber1.Synchronized = true
VideoGrabber2.Synchronized = true
VideoGrabber3.Synchronized = true
```

```
VideoGrabber1.StartPreview()
VideoGrabber2.StartPreview()
VideoGrabber3.StartPreview()
```

```
VideoGrabber1.StartSynchronized()
```

When invoking StartSynchronized on one of the 3 TVideoGrabber components (in this example VideoGrabber1) the preview will start simultaneously on the 3 components.

For the recording, replace StartPreview() by StartRecording()

## PLAYER SYNCHRONIZATION

See [Synchronization of several player components chapter](#)

---

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

---

## Logo

---

Created with the Standard Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

### Logo displayed in the video window

#### Logo displayed in the video window

[Prev](#)

[Next](#)

Logo displayed in the inactive video window.

#### Description

It is possible to fill out the inactive video window with a logo or a static image.

First of all, setup a logo by using one of the following functions:

```
SetLogoFromBMPFile (FileName: string)
SetLogoFromJPEGFile (FileName: string)
SetLogoFromHBitmap (Bitmap: HBITMAP)
SetLogoFromTBitmap (Bitmap: TBitmap)
SetLogoFromTImage (Image: TImage)
```

Then set a layout (centered, stretched, boxed, repeated, etc...) with [LogoLayout](#)

Now the logo can be shown or hidden by enabling or disabling [LogoDisplayed](#)

#### See Also

[LogoDisplayed](#) [LogoLayout](#) [SetLogoFromBMPFile](#) [SetLogoFromHBitmap](#) [SetLogoFromJPEGFile](#)  
[SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [HelpNDoc's Project Analyzer: Incredible documentation assistant](#)

## WPF

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

### VidGrabWPF:VideoGrabberWPF component

#### VidGrabWPF:VideoGrabberWPF component

[Prev](#)

[Next](#)

New VidGrabWPF:VideoGrabberWPF component

##### Description

This VidGrabWPF:VideoGrabberWPF component is a WPF component compatible with the code of the VidGrab::VideoGrabber component of WinForms.

It has a dual mode that can be switched with the VideoGrabberTolImage property listed in the "Common" properties of the component:

**"VideoGrabberTolImage" property disabled** (default):

TVideoGrabber renders in a DirectShow renderer through a WindowsFormHost.

This saves CPU, but with the potential "airspace issue" if using WPF semi-transparent overlays

**"VideoGrabberTolImage" property enabled:**

TVideoGrabber renders through an Image component, allowing WPF semi-transparent overlays

**The sample code can be found in the "MainDemoWPF" project included in the package under**

**CSharp\_VB.NET**

This demo is a WPF demo project similar to the Winform MainDemo project.

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## TVideoGrabber

#### TVideoGrabber

[Prev](#)

[Next](#)

[Properties](#) [Methods](#) [Events](#)

TVideoGrabber component.

##### Unit

[VidGrab](#)

##### Description

TVideoGrabber component events, methods and properties.

Created with the Standard Edition of HelpNDoc: [Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion](#)

## Properties

#### TVideoGrabber Properties

[TVideoGrabber](#)

**Public**

[AdjustOverlayAspectRatio](#)  
[AdjustPixelAspectRatio](#)  
[Aero](#)  
[Align](#)  
[Alignment](#)  
[AnalogVideoStandard](#)  
[AnalogVideoStandards](#)  
[AnalogVideoStandardsCount](#)  
[Anchors](#)  
[ApplicationPriority](#)  
[ASFAudioBitRate](#)  
[ASFAudioChannels](#)  
[ASFBufferWindow](#)  
[ASFDeinterlaceMode](#)  
[ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#)  
[ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#)  
[ASFNetworkMaxUsers](#)  
[ASFNetworkPort](#)  
[ASFProfile](#)  
[ASFProfileFromCustomFile](#)  
[ASFProfiles](#)  
[ASFProfilesCount](#)  
[ASFProfileVersion](#)  
[ASFVideoBitRate](#)  
[ASFVideoFrameRate](#)  
[ASFVideoHeight](#)  
[ASFVideoMaxKeyFrameSpacing](#)  
[ASFVideoQuality](#)  
[ASFVideoWidth](#)  
[AspectRatioToUse](#)  
[AssociateAudioAndVideoDevices](#)  
[AudioBalance](#)  
[AudioChannelRenderMode](#)  
[AudioCompressor](#)  
[AudioCompressorName](#)  
[AudioCompressors](#)  
[AudioCompressorsCount](#)  
[AudioDevice](#)  
[AudioDeviceName](#)  
[AudioDeviceRendering](#)  
[AudioDevices](#)  
[AudioDevicesCount](#)  
[AudioFormat](#)  
[AudioFormats](#)  
[AudioInput](#)  
[AudioInputBalance](#)  
[AudioInputLevel](#)  
[AudioInputMono](#)  
[AudioInputs](#)  
[AudioInputsCount](#)  
[AudioPeakEvent](#)  
[AudioRecording](#)  
[AudioRenderer](#)  
[AudioRendererName](#)  
[AudioRenderers](#)  
[AudioRenderersCount](#)  
[AudioSource](#)  
[AudioStreamNumber](#)  
[AudioSyncAdjustment](#)

[AudioSyncAdjustmentEnabled](#)  
[AudioVolume](#)  
[AutoFileName](#)  
[AutoFileNameDateTimeFormat](#)  
[AutoFileNameMinDigits](#)  
[AutoFilePrefix](#)  
[AutoRefreshPreview](#)  
[AutoStartPlayer](#)  
[AVIDurationUpdated](#)  
[AVIFormatOpenDML](#)  
[AVIFormatOpenDMLCompatibilityIndex](#)  
[BackgroundColor](#)  
[BorderStyle](#)  
[BurstCount](#)  
[BurstInterval](#)  
[BurstMode](#)  
[BurstType](#)  
[Busy](#)  
[BusyCursor](#)  
[CameraControlSettings](#)  
[Caption](#)  
[CaptureFileExt](#)  
[Color](#)  
[ColorKey](#)  
[ColorKeyEnabled](#)  
[CompressionMode](#)  
[CompressionType](#)  
[Cropping\\_Enabled](#)  
[Cropping\\_Height](#)  
[Cropping\\_Outbounds](#)  
[Cropping\\_Width](#)  
[Cropping\\_X](#)  
[Cropping\\_XMax](#)  
[Cropping\\_Y](#)  
[Cropping\\_YMax](#)  
[Cropping\\_Zoom](#)  
[CurrentFrameRate](#)  
[CurrentState](#)  
[DeliveredFrames](#)  
[DirectShowFilters](#)  
[DirectShowFiltersCount](#)  
[Display\\_Active](#)  
[Display\\_AlphaBlendEnabled](#)  
[Display\\_AlphaBlendValue](#)  
[Display\\_AspectRatio](#)  
[Display\\_AutoSize](#)  
[Display\\_Embedded](#)  
[Display\\_FullScreen](#)  
[Display\\_Height](#)  
[Display\\_Left](#)  
[Display\\_Monitor](#)  
[Display\\_MouseMovesWindow](#)  
[Display\\_PanScanRatio](#)  
[Display\\_StayOnTop](#)  
[Display\\_Top](#)  
[Display\\_TransparentColorEnabled](#)  
[Display\\_TransparentColorValue](#)  
[Display\\_VideoHeight](#)  
[Display\\_VideoPortEnabled](#)  
[Display\\_VideoWidth](#)  
[Display\\_VideoWindowHandle](#)  
[Display\\_Visible](#)  
[Display\\_Width](#)



[DroppedFrames](#)  
[DroppedFramesPollingInterval](#)  
[DualDisplay\\_Active](#)  
[DualDisplay\\_AlphaBlendEnabled](#)  
[DualDisplay\\_AlphaBlendValue](#)  
[DualDisplay\\_AspectRatio](#)  
[DualDisplay\\_AutoSize](#)  
[DualDisplay\\_Embedded](#)  
[DualDisplay\\_FullScreen](#)  
[DualDisplay\\_Height](#)  
[DualDisplay\\_Left](#)  
[DualDisplay\\_Monitor](#)  
[DualDisplay\\_MouseMovesWindow](#)  
[DualDisplay\\_PanScanRatio](#)  
[DualDisplay\\_StayOnTop](#)  
[DualDisplay\\_Top](#)  
[DualDisplay\\_TransparentColorEnabled](#)  
[DualDisplay\\_TransparentColorValue](#)  
[DualDisplay\\_VideoHeight](#)  
[DualDisplay\\_VideoPortEnabled](#)  
[DualDisplay\\_VideoWidth](#)  
[DualDisplay\\_VideoWindowHandle](#)  
[DualDisplay\\_Visible](#)  
[DualDisplay\\_Width](#)  
[DVDDateTimeEnabled](#)  
[DVDDiscontinuityMinimumInterval](#)  
[DVDTITLE](#)  
[DVEncoder\\_VideoFormat](#)  
[DVEncoder\\_VideoResolution](#)  
[DVEncoder\\_VideoStandard](#)  
[DVRRecordingInNativeFormatSeparatesStreams](#)  
[DVRReduceFrameRate](#)  
[DVRgb219](#)  
[DVTimeCodeEnabled](#)  
[Enabled](#)  
[EncryptionMethod](#)  
[EventNotificationSynchronise](#)  
[ExtraDLLPath](#)  
[FixFlickerOrBlackCapture](#)  
[FrameCaptureHeight](#)  
[FrameCaptureWidth](#)  
[FrameCaptureWithoutOverlay](#)  
[FrameCaptureZoomSize](#)  
[FrameGrabber](#)  
[FrameGrabberCurrentRGBFormat](#)  
[FrameGrabberRGBFormat](#)  
[FrameNumberStartsFromZero](#)  
[FrameRate](#)  
[FramerateDivider](#)  
[FullRepaint](#)  
[GetLastFrameWaitTimeoutMs](#)  
[Height](#)  
[HoldRecording](#)  
[ImageOverlay\\_AlphaBlend](#)  
[ImageOverlay\\_AlphaBlendValue](#)  
[ImageOverlay\\_ChromaKey](#)  
[ImageOverlay\\_ChromaKeyLeewayPercent](#)  
[ImageOverlay\\_ChromaKeyRGBColor](#)  
[ImageOverlay\\_Height](#)  
[ImageOverlay\\_LeftLocation](#)  
[ImageOverlay\\_RotationAngle](#)  
[ImageOverlay\\_StretchToVideoSize](#)  
[ImageOverlay\\_TopLocation](#)

[ImageOverlay\\_Transparent](#)  
[ImageOverlay\\_TransparentColorValue](#)  
[ImageOverlay\\_UseTransparentColor](#)  
[ImageOverlay\\_VideoAlignment](#)  
[ImageOverlay\\_Width](#)  
[ImageOverlayEnabled](#)  
[ImageOverlaySelector](#)  
[ImageRatio](#)  
[InFrameProgressEvent](#)  
[IPCameraURL](#)  
[IsAnalogVideoDecoderAvailable](#)  
[IsAudioCrossbarAvailable](#)  
[IsAudioDeviceASoundCard](#)  
[IsAudioInputBalanceAvailable](#)  
[IsCameraControlAvailable](#)  
[IsDigitalVideoIn](#)  
[IsDVCommandAvailable](#)  
[IsHorizontalSyncLocked](#)  
[IsMPEGStream](#)  
[IsPlayerAudioStreamAvailable](#)  
[IsPlayerVideoStreamAvailable](#)  
[IsRecordingPaused](#)  
[IsTimeCodeReaderAvailable](#)  
[IsTVAudioAvailable](#)  
[IsTVAutoTuneRunning](#)  
[IsTVTunerAvailable](#)  
[IsVideoControlAvailable](#)  
[IsVideoCrossbarAvailable](#)  
[IsVideoInterlaced](#)  
[IsVideoPortAvailable](#)  
[IsVideoQualityAvailable](#)  
[IsWDMVideoDriver](#)  
[JPEGPerformance](#)  
[JPEGProgressiveDisplay](#)  
[JPEGQuality](#)  
[Last\\_BurstFrameCapture\\_FileName](#)  
[Last\\_CaptureFrameTo\\_FileName](#)  
[Last\\_Clip\\_Played](#)  
[Last\\_Recording\\_FileName](#)  
[Left](#)  
[LogoDisplayed](#)  
[LogoLayout](#)  
[MixAudioSamples\\_CurrentSourceLevel](#)  
[MixAudioSamples\\_ExternalSourceLevel](#)  
[Mixer\\_MosaicColumns](#)  
[Mixer\\_MosaicLines](#)  
[MotionDetector\\_CompareBlue](#)  
[MotionDetector\\_CompareGreen](#)  
[MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#)  
[MotionDetector\\_GlobalMotionRatio](#)  
[MotionDetector\\_GreyScale](#)  
[MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount](#)  
[MotionDetector\\_GridYCount](#)  
[MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#)  
[MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#)  
[MotionDetector\\_Triggered](#)  
[MouseWheelEventEnabled](#)  
[MpegStreamType](#)  
[MultiplexedInputEmulation](#)

[MultiplexedRole](#)  
[MultiplexedStabilizationDelay](#)  
[MultiplexedSwitchDelay](#)  
[MuteAudioRendering](#)  
[NDIFormatType](#)  
[NDIName](#)  
[NetworkStreaming](#)  
[NetworkStreamingType](#)  
[NewProperty2](#)  
[NormalCursor](#)  
[NotificationMethod](#)  
[NotificationPriority](#)  
[NotificationSleepTime](#)  
[OnFrameBitmapEventSynchronone](#)  
[OpenURLAsync](#)  
[OverlayAfterTransform](#)  
[ParentColor](#)  
[ParentShowHint](#)  
[ParentWindow](#)  
[PlayerAudioCodec](#)  
[PlayerAudioRendering](#)  
[PlayerDuration](#)  
[PlayerDVSize](#)  
[PlayerFastSeekSpeedRatio](#)  
[PlayerFileName](#)  
[PlayerForcedCodec](#)  
[PlayerFrameCount](#)  
[PlayerFramePosition](#)  
[PlayerFrameRate](#)  
[PlayerHwAccel](#)  
[PlayerOpenProgressPercent](#)  
[PlayerRefreshPausedDisplay](#)  
[PlayerRefreshPausedDisplayFrameRate](#)  
[PlayerSpeedRatio](#)  
[PlayerSpeedRatioConstantAudioPitch](#)  
[PlayerState](#)  
[PlayerTimePosition](#)  
[PlayerTrackBar](#)  
[PlayerTrackBarScale](#)  
[PlayerTrackBarSynchronone](#)  
[PlayerVideoCodec](#)  
[PlaylistIndex](#)  
[PopupMenu](#)  
[PreallocCapFileCopiedAfterRecording](#)  
[PreallocCapFileEnabled](#)  
[PreallocCapFileName](#)  
[PreallocCapFileSizeInMB](#)  
[PreviewZoomSize](#)  
[RawAudioSampleCapture](#)  
[RawCaptureAsyncEvent](#)  
[RawSampleCaptureLocation](#)  
[RawVideoSampleCapture](#)  
[RecordingAudioBitRate](#)  
[RecordingBacktimedFramesCount](#)  
[RecordingCanPause](#)  
[RecordingDuration](#)  
[RecordingFileName](#)  
[RecordingFileSizeMaxInMB](#)  
[RecordingFourCC](#)  
[RecordingHeight](#)  
[RecordingInNativeFormat](#)  
[RecordingMethod](#)  
[RecordingOnMotion\\_Enabled](#)

[RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)  
[RecordingPauseCreatesNewFile](#)  
[RecordingSize](#)  
[RecordingTimer](#)  
[RecordingTimerInterval](#)  
[RecordingVideoBitRate](#)  
[RecordingWidth](#)  
[Reencoding\\_IncludeAudioStream](#)  
[Reencoding\\_IncludeVideoStream](#)  
[Reencoding\\_Method](#)  
[Reencoding\\_NewVideoClip](#)  
[Reencoding\\_SourceVideoClip](#)  
[Reencoding\\_StartFrame](#)  
[Reencoding\\_StartTime](#)  
[Reencoding\\_StopFrame](#)  
[Reencoding\\_StopTime](#)  
[Reencoding\\_UseAudioCompressor](#)  
[Reencoding\\_UseFrameGrabber](#)  
[Reencoding\\_UseVideoCompressor](#)  
[Reencoding\\_WMVOutput](#)  
[ScreenRecordingLayeredWindows](#)  
[ScreenRecordingMonitor](#)  
[ScreenRecordingNonVisibleWindows](#)  
[ScreenRecordingSizePercent](#)  
[ScreenRecordingThroughClipboard](#)  
[ScreenRecordingWithCursor](#)  
[SendToDV\\_DeviceIndex](#)  
[ShapeOverlay](#)  
[ShapeOverlayEnabled](#)  
[SourceStream](#)  
[SpeakerBalance](#)  
[SpeakerControl](#)  
[SpeakerVolume](#)  
[StoragePath](#)  
[StoreDeviceSettingsInRegistry](#)  
[StreamingURL](#)  
[StreamInterface\\_Format](#)  
[StreamInterface\\_FrameRate](#)  
[StreamInterface\\_IsRealTime](#)  
[SyncCommands](#)  
[SynchronizationRole](#)  
[Synchronized](#)  
[SyncPreview](#)  
[SystemTempPath](#)  
[Tag](#)  
[TextOverlay\\_Align](#)  
[TextOverlay\\_AlphaBlend](#)  
[TextOverlay\\_AlphaBlendValue](#)  
[TextOverlay\\_BkColor](#)  
[TextOverlay\\_Enabled](#)  
[TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#)  
[TextOverlay\\_Left](#)  
[TextOverlay\\_Right](#)  
[TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#)  
[TextOverlay\\_Selector](#)  
[TextOverlay\\_Shadow](#)  
[TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#)  
[TextOverlay\\_String](#)  
[TextOverlay\\_Top](#)

[TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#)  
[ThirdPartyDeinterlacer](#)  
[Top](#)  
[TranslateMouseCoordinates](#)  
[TunerFrequency](#)  
[TunerMode](#)  
[TVChannel](#)  
[TVCountryCode](#)  
[TVTunerInputType](#)  
[TVUseFrequencyOverrides](#)  
[UniqueID](#)  
[UseClock](#)  
[v360\\_AspectRatio](#)  
[v360\\_Enabled](#)  
[v360\\_MasterAngle](#)  
[v360\\_MouseAction](#)  
[v360\\_MouseActionPercent](#)  
[VCRHorizontalLocking](#)  
[Version](#)  
[VideoCompression\\_DataRate](#)  
[VideoCompression\\_KeyFrameRate](#)  
[VideoCompression\\_PFramesPerKeyFrame](#)  
[VideoCompression\\_Quality](#)  
[VideoCompression\\_WindowSize](#)  
[VideoCompressor](#)  
[VideoCompressorName](#)  
[VideoCompressors](#)  
[VideoCompressorsCount](#)  
[VideoControlSettings](#)  
[VideoCursor](#)  
[VideoDevice](#)  
[VideoDeviceName](#)  
[VideoDevices](#)  
[VideoDevicesCount](#)  
[VideoDevicesId](#)  
[VideoDoubleBuffered](#)  
[VideoFormat](#)  
[VideoFormats](#)  
[VideoFormatsCount](#)  
[VideoFromImages\\_BitmapsSortedBy](#)  
[VideoFromImages\\_RepeatIndefinitely](#)  
[VideoFromImages\\_SourceDirectory](#)  
[VideoFromImages\\_TemporaryFile](#)  
[VideoHeight](#)  
[VideoHeight\\_PreferedAspectRatio](#)  
[VideoInput](#)  
[VideoInputs](#)  
[VideoInputsCount](#)  
[VideoPlayableWhileRecording](#)  
[VideoProcessing\\_Brightness](#)  
[VideoProcessing\\_Contrast](#)  
[VideoProcessing\\_Deinterlacing](#)  
[VideoProcessing\\_FlipHorizontal](#)  
[VideoProcessing\\_FlipVertical](#)  
[VideoProcessing\\_GrayScale](#)  
[VideoProcessing\\_Hue](#)  
[VideoProcessing\\_InvertColors](#)  
[VideoProcessing\\_Pixellization](#)  
[VideoProcessing\\_Rotation](#)  
[VideoProcessing\\_RotationCustomAngle](#)  
[VideoProcessing\\_Saturation](#)  
[VideoQualitySettings](#)

[VideoRenderer](#)  
[VideoRendererExternal](#)  
[VideoRendererExternalIndex](#)  
[VideoRendererPriority](#)  
[VideoSize](#)  
[VideoSizes](#)  
[VideoSizesCount](#)  
[VideoSource](#)  
[VideoSource\\_FileOrURL](#)  
[VideoSource\\_FileOrURL\\_StartTime](#)  
[VideoSource\\_FileOrURL\\_StopTime](#)  
[VideoSources](#)  
[VideoSourcesCount](#)  
[VideoStreamNumber](#)  
[VideoSubtype](#)  
[VideoSubtypes](#)  
[VideoSubtypesCount](#)  
[VideoVisibleWhenStopped](#)  
[VideoWidth](#)  
[VideoWidth\\_PreferedAspectRatio](#)  
[Visible](#)  
[VUMeter](#)  
[WebcamStillCaptureButton](#)  
[Width](#)  
[ZoomCoeff](#)  
[ZoomXCenter](#)  
[ZoomYCenter](#)

Created with the Standard Edition of HelpNDoc: [Protect Your Confidential PDFs with These Simple Security Measures](#)

## AdjustOverlayAspectRatio

### TVideoGrabber.AdjustOverlayAspectRatio

[Next](#)

[TVideoGrabber](#) [Properties](#)  
[ber](#)

Adjusts the aspect ratio of the video frames before applying the overlays

#### Declaration

**property** AdjustOverlayAspectRatio: Boolean **read** GetAdjustOverlayAspectRatio **write** SetAdjustOverlayAspectRatio **default** DEF\_AdjustOverlayAspectRatio;

**\_\_property bool** AdjustOverlayAspectRatio=read=GetAdjustOverlayAspectRatio, write=SetAdjustOverlayAspectRatio, **default**=1

Property AdjustOverlayAspectRatio As Boolean

#### Description

Used to adjust the aspect ratio of the video frames before applying the overlays.

This property applies only when the aspect ratio of the video source does not match the size of the video frames (e.g. a 16x9 aspect ratio for a 720x480 native video size)

#### AdjustOverlayAspectRatio enabled (requires a few CPU more):

The video frames are resized to their display aspect ratio **before** the overlays are applied, so the aspect ratio of the overlays is correct.

#### AdjustOverlayAspectRatio disabled (saves CPU):

The video frames are not resized, the overlays are applied to the native video size.

If the video is displayed to its correct aspect ratio (e.g. when [Display AspectRatio](#) = ar\_Box), the video frames will be stretched, including the overlays that may appear distorted.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

## AdjustPixelAspectRatio

### TVideoGrabber.AdjustPixelAspectRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enable/disable the correction of non-square pixels

#### Declaration

**property** AdjustPixelAspectRatio: Boolean **read** GetAdjustPixelAspectRatio **write** SetAdjustPixelAspectRatio  
**default** DEF\_AdjustPixelAspectRatio;

**\_\_property bool** AdjustPixelAspectRatio=read=GetAdjustPixelAspectRatio,  
write=SetAdjustPixelAspectRatio, **default=1**

Property AdjustPixelAspectRatio As Boolean

#### Description

Used to enable / disable the correction of non-square pixels, e.g. when the video source is a PAL source or a NTSC source.

When [AdjustPixelAspectRatio](#) is enabled (by default) the pixel aspect ratio is corrected when a PAL or NTSC format is detected, and also according to the monitor aspect ratio. This concerns the video window and the frame capture size.

When [AdjustPixelAspectRatio](#) is disabled, the video frame will be processed "as is", but it can be stretched to the size of the video window if [Display AutoSize](#) = false.

To display the video AS IS (without stretching the video frame to the video window size):  
- if you want the control to be resized automatically, set [AdjustPixelAspectRatio](#) = false **and**

[Display\\_AutoSize](#) = true

- if you want to specify yourself the size of the control, set [AdjustPixelAspectRatio](#) = false **and**

[Display\\_AutoSize](#) = false **and** [Display\\_AspectRatio](#) = ar\_NoResize

#### See Also

[TVideoRenderer](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

## Aero

### TVideoGrabber.Aero

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Temporarily disables/enables the Aero mode of Vista/Windows 7

#### Declaration

**property** Aero: TAero **read** GetAero **write** SetAero **default** DEF\_Aero;

**\_\_property** TAero Aero=read=GetAero, write=SetAero, **default**=1

Property Aero As TAero

#### Description

This [TAero](#) property is used to disable/enable the Windows Aero mode (the transparency of the window titles in Vista / Windows 7)

By default the Windows Vista and Windows 7 desktop use the "Aero" mode (in this mode you can see the background under the transparent borders and title of the windows).

In this mode the overlay rendering can't be used.

To get a better video quality when playing MPEG clips in full screen mode with the VMR7 / VMR9 and the frame grabber is disabled, you can temporarily disable the Aero mode then the video starts playing by setting Aero = **ae\_ForceOffWhenStartingVideo**

If you prefer to disable the Aero mode immediately set Aero = ae\_ForceOffImmediately (then set Aero = ae\_ForceOnImmediately to restore it).

*Note that it is not possible to enable the Aero mode if it was disabled when starting the application (unless changing the properties of the desktop), it is only possible to disable the Aero mode if it was enabled, then re-enable it and re-disable as needed.*

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)



## AnalogVideoStandard

### TVideoGrabber.AnalogVideoStandard

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects an analog video standard.

#### Declaration

**property** AnalogVideoStandard: LongInt **read** GetAnalogVideoStandard **write** SetAnalogVideoStandard;  
 \_\_property int AnalogVideoStandard=read=GetAnalogVideoStandard, write=SetAnalogVideoStandard,  
**nodefault**

Property AnalogVideoStandard As Long

#### Description

Used to select an analog video standard for the [current video capture device](#) in the [AnalogVideoStandards](#) list.

#### Important:

The analog standard is not the same from a video capture device to the other. It is reloaded when the video capture device is selected by assigning the [VideoDevice](#) property.

- any control displaying this value should re-read it from the [OnVideoDeviceSelected](#) event (that occurs when the video device is selected)
- be sure that the video capture device has been selected (by assigning the [VideoDevice](#) property) before setting this property

See [device-dependent properties](#)

#### See Also

[Analog Video standards](#) [AnalogVideoStandardIndex](#) [AnalogVideoStandards](#) [AnalogVideoStandardsCount](#) [IsAnalogVideoDecoderAvailable](#)

Created with the Standard Edition of HelpNDoc: [Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

## AnalogVideoStandards

### TVideoGrabber.AnalogVideoStandards

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of analog video standards available for the current video capture device.

#### Declaration

**property** AnalogVideoStandards: **string read** GetAnalogVideoStandards;  
 \_\_property wchar\_t \*AnalogVideoStandards=read=GetAnalogVideoStandards

Property AnalogVideoStandards As String

#### Description

Used to retrieve a string that contains the list of the video standards available on the current video capture device when when [IsAnalogVideoDecoderAvailable](#) returns true.

The [AnalogVideoStandard](#) property is an index in this list, used to select the current analog video standard.

This list is updated when the [OnVideoDeviceSelected](#) event occurs (when a video capture device is

selected with [VideoDevice](#) ).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.AnalogVideoStandards*

*ComboBox1.ItemIndex := VideoGrabber1.AnalogVideoStandard*

#### See Also

[Analog Video standards](#) [AnalogVideoStandard](#) [AnalogVideoStandardIndex](#) [AnalogVideoStandardsCount](#)  
[IsAnalogVideoDecoderAvailable](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## AnalogVideoStandardsCount

### TVideoGrabber.AnalogVideoStandardsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of analog video standards.

#### Declaration

**property** AnalogVideoStandardsCount: LongInt **read** GetAnalogVideoStandardsCount;

\_\_property **int** AnalogVideoStandardsCount=read=GetAnalogVideoStandardsCount, **nodefault**

Property AnalogVideoStandardsCount As Long

#### Description

Number of analog video standards in the [AnalogVideoStandards](#) list for the current video capture device.

The [AnalogVideoStandard](#) property is an index used to select the current analog video standard, in the **0 ... AnalogVideoStandardsCount - 1** range.

#### See Also

[Analog Video standards](#) [AnalogVideoStandard](#) [AnalogVideoStandardIndex](#) [AnalogVideoStandards](#)  
[IsAnalogVideoDecoderAvailable](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your PDF Protection with These Simple Steps](#)

## ApplicationPriority

### TVideoGrabber.ApplicationPriority

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets the priority of the application.

#### Declaration

**property** ApplicationPriority: TApplicationPriority **read** GetApplicationPriority **write** SetApplicationPriority  
**default** DEF\_ApplicationPriority;

\_\_property TApplicationPriority ApplicationPriority=read=GetApplicationPriority,  
write=SetApplicationPriority, **default**=0

#### Description

Used to set the priority of the application.

<b>Values</b>	normal priority	idle priority	normal priority	high priority	real time priority
ap_default:	ap_idle:	ap_normal:	ap_high:	ap_realtime:	
					Although it looks like threading priority, this property is used to select the

accuracy of the frame counting by the [OnFrameProgress2](#) event.

It is recommend to leave it to its default value.

Setting it to `ap_idle` may reduce the CPU load but may not report all the frames.

Created with the Standard Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

## ASFAudioBitRate

### TVideoGrabber.ASFAudioBitRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the bit rate of the audio stream in bits per second.

#### Declaration

**property** ASFAudioBitRate: LongInt **read** GetASFAudioBitRate **write** SetASFAudioBitRate **default** DEF\_ASFAudioBitRate;

\_\_property **int** ASFAudioBitRate=read=GetASFAudioBitRate, write=SetASFAudioBitRate, **default**=- 1

Property ASFAudioBitRate As Long

#### Description

Specifies the bit rate of the audio stream in bits per second.

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile is used.

E.g. a value of 40000 specifies about 40 Kbits per second.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Documentation Workflow with HelpNDoc's Intuitive UI](#)

## ASFAudioChannels

### TVideoGrabber.ASFAudioChannels

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

## ber

Specifies the number of channels of the audio stream.

### Declaration

**property** ASFAudioChannels: LongInt **read** GetASFAudioChannels **write** SetASFAudioChannels **default** DEF\_ASFAudioChannels;

\_\_property int ASFAudioChannels=read=GetASFAudioChannels, write=SetASFAudioChannels, **default**=- 1

Property ASFAudioChannels As Long

### Description

Specifies the number of channels of the audio stream.

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile is used.

1: 1 channel (mono)

2: 2 channels (stereo)

### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## ASFBufferWindow

### TVideoGrabber.ASFBufferWindow

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Increases the streaming latency

### Declaration

**property** ASFBufferWindow: LongInt **read** GetASFBufferWindow **write** SetASFBufferWindow **default** DEF\_ASFBufferWindow;

\_\_property int ASFBufferWindow=read=GetASFBufferWindow, write=SetASFBufferWindow, **default**=- 1

Property ASFBufferWindow As Long

### Description

Used to increase the streaming latency, expressed in milliseconds.

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## ASFDeinterlaceMode

### TVideoGrabber.ASFDeinterlaceMode

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

**ber**

Specifies if the video stream must be deinterlaced.

**Declaration**

**property** ASFDeinterlaceMode: TASFDeinterlaceMode **read** GetASFDeinterlaceMode **write** SetASFDeinterlaceMode **default** DEF\_ASFDeinterlaceMode;

\_\_property TASFDeinterlaceMode ASFDeinterlaceMode=read=GetASFDeinterlaceMode, write=SetASFDeinterlaceMode, **default**=0

Property ASFDeinterlaceMode As TxASFDeinterlaceMode

**Description**

Specifies if the video stream must be deinterlaced before being recorded.  
The value is chosen in the [TASFDeinterlaceMode](#) type.

**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion

**ASFFixedFrameRate****TVideoGrabber.ASFFixedFrameRate**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Forces the ASF recording to use a fixed frame rate.

**Declaration**

**property** ASFFixedFrameRate: Boolean **read** GetASFFixedFrameRate **write** SetASFFixedFrameRate **default** DEF\_ASFFixedFrameRate;

\_\_property **bool** ASFFixedFrameRate=read=GetASFFixedFrameRate, write=SetASFFixedFrameRate, **default**=0

Property ASFFixedFrameRate As Boolean

**Description**

Specifies if a fixed frame rate must be used to record the video stream.  
By default the ASF frame rate is not constant.

**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#)

[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## ASFMediaServerPublishingPoint

### TVideoGrabber.ASFMediaServerPublishingPoint

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies a publishing point on a Windows Media Server.

#### Declaration

**property** ASFMediaServerPublishingPoint: **string read** GetASFMediaServerPublishingPoint **write** SetASFMediaServerPublishingPoint;

\_\_property wchar\_t \*ASFMediaServerPublishingPoint=read=GetASFMediaServerPublishingPoint, write=SetASFMediaServerPublishingPoint

Property ASFMediaServerPublishingPoint As String

#### Description

Used to specify a publishing point on a Windows Media Server, to which the audio/video stream will be sent.

This property is used when [Networking](#) = ns\_ASFStreamingToPublishingPoint.

E.g.: http://your\_WM\_server.com/your\_publishing\_point

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

## ASFMediaServerRemovePublishingPointAfterDisconnect

### TVideoGrabber.ASFMediaServerRemovePublishingPointAfterDisconnect

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies if a publishing point created dynamically must be destroyed.

#### Declaration

**property** ASFMediaServerTemplatePublishingPoint: **string read** GetASFMediaServerTemplatePublishingPoint **write** SetASFMediaServerTemplatePublishingPoint;

\_\_property wchar\_t \*ASFMediaServerPublishingPoint=read=GetASFMediaServerTemplatePublishingPoint,

write=SetASFMediaServerTemplatePublishingPoint

Property ASFMediaServerTemplatePublishingPoint As String

### Description

Used to specify if the publishing point created dynamically on the Windows Media server must be destroyed when disconnecting.

### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Full-featured EPub generator

## ASFMediaServerTemplatePublishingPoint

**TVideoGrabber.ASFMediaServerTemplatePublishingPoint**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies a template publishing point on a Windows Media Server.

### Declaration

**property** ASFMediaServerRemovePublishingPointAfterDisconnect: **boolean** **read**  
GetASFMediaServerRemovePublishingPointAfterDisconnect **write**  
SetASFMediaServerRemovePublishingPointAfterDisconnect;

\_\_property bool

ASFMediaServerRemovePublishingPointAfterDisconnect=read=GetASFMediaServerRemovePublishingPointAfterDisconnect, write=SetASFMediaServerRemovePublishingPointAfterDisconnect

Property ASFMediaServerRemovePublishingPointAfterDisconnect As Bool

### Description

Used to specify the name of a template publishing point on the Windows Media Server, that will be used to create automatically the publishing point specified by [ASFMediaServerPublishingPoint](#) .

This property is used when [Networking](#) = ns\_ASFStreamingToPublishingPoint.

### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Experience the Power and Ease of Use of HelpNDoc for



## ASFNetworkMaxUsers

### TVideoGrabber.ASFNetworkMaxUsers

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the maximum number of users for direct network streaming.

#### Declaration

**property** ASFNetworkMaxUsers: LongInt **read** GetASFNetworkMaxUsers **write** SetASFNetworkMaxUsers  
**default** DEF\_ASFNetworkMaxUsers;

\_\_property int ASFNetworkMaxUsers=read=GetASFNetworkMaxUsers, write=SetASFNetworkMaxUsers,  
**default**=5

Property ASFNetworkMaxUsers As Long

#### Description

Used to specify the maximum number of users that can connect to the current platform when network streaming is enabled by [Networking](#) = ns\_ASFDirectNetworkStreaming

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#)  
[ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#)  
[ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#)  
[ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#)  
[OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation with HelpNDoc's Project Analyzer Features](#)

## ASFNetworkPort

### TVideoGrabber.ASFNetworkPort

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the network port for direct network streaming.

#### Declaration

**property** ASFNetworkPort: LongInt **read** GetASFNetworkPort **write** SetASFNetworkPort **default**  
DEF\_ASFNetworkPort;

\_\_property int ASFNetworkPort=read=GetASFNetworkPort, write=SetASFNetworkPort, **default**=0

Property ASFNetworkPort As Long

#### Description

Used to specify the network port used to perform network streaming from the when network streaming is enabled by [Networking](#) = ns\_ASFDirectNetworkStreaming

If this property is set to 0, a valid port number is automatically assigned and returned by the [OnDirectNetworkStreamingHostUrl](#) event.



**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFProfile](#) [ASFProfileFromCustomFile](#)  
[ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#)  
[ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#)  
[ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#)  
[OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Write EPub books for the iPad

**ASFProfile****TVideoGrabber.ASFProfile**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Index of the current ASF profile.

**Declaration**

**property** ASFProfile: Integer **read** GetASFProfile **write** SetASFProfile **default** DEF\_ASFProfile;

\_\_property **int** ASFProfile=read=GetASFProfile, write=SetASFProfile, **default=-1**

Property ASFProfile As Long

**Description**

Index of the current ASF profile in the [ASFProfiles](#) list.

Set this value to **-1** to disable ASF profiles and use default ASF streaming values.

**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#)  
[ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#)  
[ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#)  
[ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#)  
[NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#)  
[ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Transform Your Documentation Workflow with HelpNDoc's Intuitive UI

**ASFProfileFromCustomFile****TVideoGrabber.ASFProfileFromCustomFile**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects a file that contains a custom profile

**Declaration**

**property** ASFProfileFromCustomFile: **string** **read** GetASFProfileFromCustomFile **write** SetASFProfileFromCustomFile;

```
__property wchar_t *ASFProfileFromCustomFile=read=GetASFProfileFromCustomFile,
write=SetASFProfileFromCustomFile
```

Property ASFProfileFromCustomFile As String

### Description

Used to select a file that contains a custom profile (usually a .prx file).

When a file is assigned to this property, the [ASFProfile](#) property is ignored and this profile is used instead.

E.g. ASFProfileFromCustomFile = "myprofile.prx"

### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfiles](#)  
[ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#)  
[ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#)  
[ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#)  
[OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Free Kindle producer

## ASFProfiles

### TVideoGrabber.ASFProfiles

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the ASF profiles available on the current platform.

### Declaration

**property** ASFProfiles: **String** read GetASFProfiles;

```
__property wchar_t *ASFProfiles=read=GetASFProfiles
```

Property ASFProfiles As String

### Description

Used to retrieve the ASF profiles available on the current platform.

The [ASFProfile](#) property is the index in this list of the current ASF profile.

Selecting an ASF profile let you choose the streaming format (video size, bit rate, etc...).

This list can be assigned to a TStringList, e.g. :

```
...
StringList := TStringList.Create;
StringList.Text := VideoGrabber.ASFProfiles;
...
```

### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#)  
[ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#)  
[ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#)

[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Qt Help documentation made easy](#)

## ASFProfilesCount

### TVideoGrabber.ASFProfilesCount

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of ASF profiles available on the current platform.

#### Declaration

**property** ASFProfilesCount: LongInt **read** GetASFProfilesCount;

\_\_property **int** ASFProfilesCount=read=GetASFProfilesCount, **nodefault**

Property ASFProfilesCount As Long

#### Description

Used to retrieve the number of ASF profiles available on the current platform.  
The [ASFProfile](#) index must be in the 0...[ASFProfilesCount](#) range.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#) [ASFStreaming\\_GetConnectedClientsCount](#) [ASFStreaming\\_ResetAuthorizations](#) [ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## ASFProfileVersion

### TVideoGrabber.ASFProfileVersion

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the Windows Media Format mode (WMV8 or WMV9).

#### Declaration

**property** ASFProfileVersion: TASFProfileVersion **read** GetASFProfileVersion **write** SetASFProfileVersion **default** DEF\_ASFProfileVersion;

\_\_property TASFProfileVersion ASFProfileVersion==GetASFProfileVersion, write=SetASFProfileVersion, default=0;

property ASFProfileVersion as TASFProfileVersion

#### Description

Used to specify the Windows Media profiles version (8 or 9), that determines the Windows Media Format mode used (WMV8 or WMV9).

apv\_ProfileVersion\_8: uses the Windows Media 8 encoders and profiles

apv\_ProfileVersion\_9: uses the Windows Media 9 encoders and profiles

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFStreaming\\_GetAuthorizationList](#)  
[ASFStreaming\\_GetConnectedClients](#) [ASFStreaming\\_GetConnectedClientsCount](#)  
[ASFStreaming\\_ResetAuthorizations](#) [ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#)  
[OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Revolutionize Your Documentation Review with  
 HelpNDoc's Project Analyzer

## ASFVideoBitRate

### TVideoGrabber.ASFVideoBitRate

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the bit rate of the video stream in bits per second.

#### Declaration

**property** ASFVideoBitRate: LongInt **read** GetASFVideoBitRate **write** SetASFVideoBitRate **default**  
 DEF\_ASFVideoBitRate;

\_\_**property** int ASFVideoBitRate=read=GetASFVideoBitRate, write=SetASFVideoBitRate, **default**=- 1

Property ASFVideoBitRate As Long

#### Description

Specifies the bit rate of the video stream in bits per second.

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile is used.

E.g. a value of 185000 specifies about 185 Kbits per second.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#)  
[ASFStreaming\\_GetConnectedClientsCount](#) [ASFStreaming\\_ResetAuthorizations](#)  
[ASFStreaming\\_SetAuthorization](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#)  
[ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#)  
[OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Make Your PDFs More Secure with Encryption and  
 Password Protection

## ASFVideoFrameRate

### TVideoGrabber.ASFVideoFrameRate

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the frame rate of the video stream in frames per second.

**Declaration**

**property** ASFVideoFrameRate: Double **read** GetASFVideoFrameRate **write** SetASFVideoFrameRate  
**default** DEF\_ASFVideoFrameRate;

\_\_**property double** ASFVideoFrameRate=read=GetASFVideoFrameRate, write=SetASFVideoFrameRate,  
**default=- 1**

Property ASFVideoFrameRate As Double

**Description**

Specifies the frame rate of the video stream expressed in frames per second.

If the specified value is -1, the frame rate of the video source is used, or the current [ASF profile](#) value if an ASF profile is used.

E.g. a value of 29.97 specifies 29.97 frames per second

**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#)  
[ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#)  
[ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#)  
[ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#)  
[OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: iPhone web sites made easy

**ASFVideoHeight****TVideoGrabber.ASFVideoHeight**

[Prev](#)

[Next](#)

[TVideoGrabber](#) **Properties**

Video height for the ASF video stream.

**Declaration**

**property** ASFVideoHeight: LongInt **read** GetASFVideoHeight **write** SetASFVideoHeight **default**  
DEF\_ASFVideoHeight;

\_\_**property int** ASFVideoHeight=read=GetASFVideoHeight, write=SetASFVideoHeight, **default=240**

Property ASFVideoHeight As Long

**Description**

Specifies the video height for the ASF video stream.

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile is used.

**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#)  
[ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#)  
[ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoMaxKeyFrameSpacing](#)  
[ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#)

[OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Easily create CHM Help documents](#)

## ASFVideoMaxKeyFrameSpacing

### TVideoGrabber.ASFVideoMaxKeyFrameSpacing

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Maximum interval between key frames.

#### Declaration

**property** ASFVideoMaxKeyFrameSpacing: LongInt **read** GetASFVideoMaxKeyFrameSpacing **write** SetASFVideoMaxKeyFrameSpacing **default** DEF\_ASFVideoMaxKeyFrameSpacing;

\_\_property **int** ASFVideoMaxKeyFrameSpacing=read=GetASFVideoMaxKeyFrameSpacing, write=SetASFVideoMaxKeyFrameSpacing, **default**=- 1

Property ASFVideoMaxKeyFrameSpacing As Long

#### Description

Specifies the maximum interval between key frames for the current ASF video stream.

This is the maximum key-frame spacing in 100-nanosecond units.

E.g. 20000 = 1 key frame every 2 seconds.

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile is used.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFStreaming](#) [GetAuthorizationList](#) [ASFStreaming](#) [GetConnectedClients](#) [ASFStreaming](#) [GetConnectedClientsCount](#) [ASFStreaming](#) [ResetAuthorizations](#) [ASFStreaming](#) [SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

## ASFVideoQuality

### TVideoGrabber.ASFVideoQuality

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Quality setting for the video stream.

#### Declaration

**property** ASFVideoQuality: LongInt **read** GetASFVideoQuality **write** SetASFVideoQuality **default** DEF\_ASFVideoQuality;

\_\_property **int** ASFVideoQuality=read=GetASFVideoQuality, write=SetASFVideoQuality, **default**=- 1

Property ASFVideoQuality As Long

#### Description

Specifies the quality setting for the video stream, in the 0 ... 100 range (100 = maximum quality).

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile



is used.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#)  
[ASFStreaming\\_GetConnectedClientsCount](#) [ASFStreaming\\_ResetAuthorizations](#)  
[ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#)  
[ASFVideoMaxKeyFrameSpacing](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#)  
[OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#)  
[StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Effortlessly create a professional-quality documentation website with HelpNDoc

## ASFVideoWidth

### TVideoGrabber.ASFVideoWidth

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Video width for the ASF video stream.

#### Declaration

**property** ASFVideoWidth: LongInt **read** GetASFVideoWidth **write** SetASFVideoWidth **default** DEF\_ASFVideoQuality;

\_\_property **int** ASFVideoWidth=read=GetASFVideoWidth, write=SetASFVideoWidth, **default**=- 1

Property ASFVideoWidth As Long

#### Description

Specifies the video width for the ASF video stream.

If the specified value is -1, the default ASF value is used, or the current [ASF profile](#) value if an ASF profile is used.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#)  
[ASFStreaming\\_GetConnectedClientsCount](#) [ASFStreaming\\_ResetAuthorizations](#)  
[ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#)  
[ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [NetworkStreaming](#) [NetworkStreamingType](#)  
[OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#)  
[StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Single source CHM, PDF, DOC and HTML Help creation

## AspectRatioToUse

### TVideoGrabber.AspectRatioToUse

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Forces the aspect ratio of the video clip played

**Declaration**

**property** AspectRatioToUse: Double **read** GetAspectRatioToUse **write** SetAspectRatioToUse **default** DEF\_AspRatioToUse;

\_\_**property double** AspectRatioToUse=read=GetAspectRatioToUse, write=SetAspectRatioToUse, **default**=1

property AspectRatioToUse as Double

**Description**

Used to specify an aspect ratio for the video clip that will be played.

By default the proper aspect ratio is applied when the [AdjustPixelAspectRatio](#) is enabled.

It can be overridden by specifying an AspectRatioToUse value > 0 (e.g. 1.3333 for 4:3, 1666667 for 16:9, etc...)

---

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

---

**AssociateAudioAndVideoDevices****TVideoGrabber.AssociateAudioAndVideoDevices**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to associate the current audio and video capture devices.

**Declaration**

**property** AssociateAudioAndVideoDevices: Boolean **read** GetAssociateAudioAndVideoDevices **write** SetAssociateAudioAndVideoDevices **default** DEF\_AssociateAudioAndVideoDevices;

\_\_**property bool** AssociateAudioAndVideoDevices=read=GetAssociateAudioAndVideoDevices, write=SetAssociateAudioAndVideoDevices, **default**=0

Property AssociateAudioAndVideoDevices As Boolean

**Description**

When enabled:

- when selecting an audio capture device by assigning a value to [AudioDevice](#), this device, the current audio states (1) are associated to the current [video capture device](#).
- therefore, when selecting later this video capture device by assigning a value to the [VideoDevice](#) property, the associated audio capture device, its audio input and the audio input values are automatically reloaded.

The audio properties associated are:

[AudioDevice](#)  
[AudioInput](#)  
[AudioInputLevel](#)  
[SpeakerBalance](#)  
[SpeakerVolume](#)

**See Also**

[TAudioFormat](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

---

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---



## AudioBalance

### TVideoGrabber.AudioBalance

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Audio left-right balance of the current [audio capture device](#) .

#### Declaration

**property** AudioBalance: LongInt **read** GetAudioBalance **write** SetAudioBalance;

\_\_property **int** AudioBalance=read=GetAudioBalance, write=SetAudioBalance, **nodefault**

Property AudioBalance As Long

#### Description

Used to set/retrieve the audio left-right balance of the current [audio capture device](#) when [AudioDeviceRendering](#) is enabled.

The valid range is from -32767 to 32767.

0 selects the center point.

**This a rendering audio property only, it does not apply to the audio recording.**

#### See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

## AudioChannelRenderMode

### TVideoGrabber.AudioChannelRenderMode

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies how the left and/or right audio channels are rendered

#### Declaration

**property** AudioChannelRenderMode: TAudioChannelRenderMode **read** GetAudioChannelRenderMode **write** SetAudioChannelRenderMode **default** acrm\_Normal;

\_\_property TAudioChannelRenderMode AudioChannelRenderMode=read=GetAudioChannelRenderMode, write=SetAudioChannelRenderMode, **default** 0

property AudioChannelRenderMode as TAudioChannelRenderMode

#### Description

Used to specify the way the right and left audio channels of the current audio stream are rendered

To be taken in account this property must be set before invoking [OpenPlayer](#), [StartPreview](#), [StartRecording](#), etc...

Possible values:

**acrm\_Normal (default)** : normal rendering of the left and right audio channels

**acrm\_RenderLeft** : the left channel is converted as "mono" and rendered on both sides (the right channel is discarded)

**acrm\_RenderRight** : the right channel is converted as "mono" and rendered on both sides (the left channel is discarded)

**acrm\_MuteLeft** : the left channel is muted

**acrm\_MuteRight** : the right channel is muted

**acrm\_Mute\_All** : both channels are muted

**acrm\_MixLeftAndRight** : the left and right channels are mixed

**acrm\_PassThru** : like acrm\_Normal, just activates the AudioChannelRenderMode processing

Remark: **acrm\_PassThru** must be used to activate the AudioChannelRenderMode processing without modifying the audio samples immediately (e.g. to be able to mute the audio later with acrm\_Mute\_All), otherwise the AudioChannelRenderMode feature will be disabled if the preview, recording or playback starts with AudioChannelRenderMode = acrm\_Normal.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TVUMeter](#) [TVUMeterSetting](#) [TOnPlayerBufferingData](#) [AudioBalance](#) [AudioStreamNumber](#) [AudioVolume](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [MuteAudioRendering](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## AudioCompressor

### TVideoGrabber.AudioCompressor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects the current audio compressor.

#### Declaration

**property** AudioCompressor: LongInt **read** GetAudioCompressor **write** SetAudioCompressor;

\_\_\_**property** int AudioCompressor=read=GetAudioCompressor, write=SetAudioCompressor, **nodefault**

Property AudioCompressor As Long

#### Description

Used to select the current audio compressor in the global [AudioCompressors](#) list.

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression\\_PFramesPerKeyFrame](#) [VideoCompression\\_Quality](#) [VideoCompression\\_WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

## AudioCompressorName

### TVideoGrabber.AudioCompressorName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Name of the current audio compressor.

#### Declaration

**property** AudioCompressorName: **string** read GetAudioCompressorName;

\_\_property wchar\_t \*AudioCompressorName=read=GetAudioCompressorName

Property AudioCompressorName As String

#### Description

Used to retrieve the name of the audio compressor selected by [AudioCompressor](#) in the global [AudioCompressors](#) list.

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression\\_PFramesPerKeyFrame](#) [VideoCompression\\_Quality](#) [VideoCompression\\_WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## AudioCompressors

### TVideoGrabber.AudioCompressors

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video compressors (audio codecs) available on the current platform.

#### Declaration

function AudioCompressors: **string**;

\_\_property wchar\_t \* \_\_fastcall AudioCompressors();

Property AudioCompressors as String

#### Description

Used to retrieve a string that contains the list of the audio compressors (audio codecs) available on the current platform.

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.AudioCompressors;*

*ComboBox1.ItemIndex := VideoGrabber1.AudioCompressor;*

It is possible to retrieve programmatically the index of a Audio compressor by using the [FindIndexInListByName](#) function as follows:

*VideoGrabber.AudioCompressor := AudioGrabber.FindIndexInListByName (VideoGrabber.AudioCompressors, VideoGrabber.AudioCompressorName);*

#### Note:

when the application starts the [VideoCompressors](#) and [AudioCompressors](#) lists are populated with the codecs currently installed. To refresh the compressor lists after installing/uninstalling codecs without

exiting/restarting your application, invoke [RefreshDevicesAndCompressorsLists](#).

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

## AudioCompressorsCount

### TVideoGrabber.AudioCompressorsCount

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of audio compressors (codecs) available on the current platform.

#### Declaration

**property** AudioCompressorsCount: LongInt **read** GetGlobal\_AudioCompressorsCount;

\_\_property **int** AudioCompressorsCount=read=GetGlobal\_AudioCompressorsCount, **nodefault**

Function AudioCompressorsCount as Long

#### Description

Number of audio compressors (codecs) in the [AudioCompressors](#) list.

**Note:** in Delphi and C++Builder versions of the component, this property is available also as a [AudioCompressorsCount](#) global variable.

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## AudioDevice

### TVideoGrabber.AudioDevice

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects the current audio capture device.

#### Declaration

**property** AudioDevice: LongInt **read** GetAudioDevice **write** SetAudioDevice **default** DEF\_AudioDevice;

\_\_property **int** AudioDevice=read=GetAudioDevice, write=SetAudioDevice, **default**=0

Property AudioDevice As Long

## Description

Used to select the current audio capture device in the [AudioDevices](#) list. When a new value is assigned to this property, the related [device-dependent values](#) are loaded from the registry and the [OnAudioDeviceSelected](#) event occurs.

It is possible to select the audio capture device programmatically by its name (as it appears in the [AudioDevices](#) list) by using the [FindIndexInListByName](#) function, e.g.:

```
procedure TfrmMainForm.Button1Click(Sender: TObject);
var
  i: LongInt;
begin
  i := VideoGrabber.FindIndexInListByName (VideoGrabber.AudioDevices, 'Realtek AC97 Au
  if i > -1 then begin // if this Device exists...
    VideoGrabber.AudioDevice := i;
  end;
end;
end;
```

## See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## AudioDeviceName

### TVideoGrabber.AudioDeviceName

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Name of the current audio capture device.

## Declaration

**property** AudioDeviceName: **string** read GetAudioDeviceName;

\_\_property wchar\_t \*AudioDeviceName=read=GetAudioDeviceName

Property AudioDeviceName As String

## Description

Used to retrieve the name of the audio capture device selected by [AudioDevice](#) in the [AudioDevices](#) list.

## See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## AudioDeviceRendering

### TVideoGrabber.AudioDeviceRendering

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

If enabled, the audio stream is rendered during preview or recording.

#### Declaration

**property** AudioDeviceRendering: Boolean **read** GetAudioDeviceRendering **write** SetAudioDeviceRendering **default** DEF\_AudioDeviceRendering;

\_\_property **bool** AudioDeviceRendering=read=GetAudioDeviceRendering, write=SetAudioDeviceRendering, **default**=0

Property AudioDeviceRendering As Boolean

#### Description

If enabled, the current [AudioDevice](#) audio capture device is added to the preview or recording graph and the audio stream is rendered on the default sound device.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## AudioDevices

### TVideoGrabber.AudioDevices

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

List of the audio capture devices available on the current platform.

#### Declaration

function AudioDevices: **string**;

\_\_property wchar\_t \* \_\_fastcall AudioDevices();

Property AudioDevices as string

#### Description

Used to retrieve a string that contains the list of audio capture devices available on the current platform. This list is updated when an audio capture device is connected or removed (when the [OnDeviceArrivalOrRemoval](#) event occurs).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := AudioDevices;*

*ComboBox1.ItemIndex := TVideoGrabber1.AudioDevice; (\* index in the AudioDevices list. \*)*

It is possible to retrieve programmatically the index of an audio capture device by using the

[FindIndexInListByName](#) function as follows:

*TVideoGrabber.AudioDevice := TVideoGrabber.FindIndexInListByName (TVideoGrabber.AudioDevices*

**Note:** in Delphi and C++Builder versions of the component, this property is available also as a

[AudioDevices](#) global variable.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## AudioDevicesCount

### TVideoGrabber.AudioDevicesCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of audio capture devices available on the current platform.

#### Declaration

**property** AudioDevicesCount: LongInt **read** GetGlobal\_AudioDevicesCount;

\_\_property **int** AudioDevicesCount=read=GetGlobal\_AudioDevicesCount, **nodefault**

Function AudioDevicesCount as Long

#### Description

Number of audio capture devices in the [AudioDevices](#) list.

**Note:** in Delphi and C++Builder versions of the component, this property is available also as a [AudioDevicesCount](#) global variable.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

## AudioFormat

### TVideoGrabber.AudioFormat

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects an audio format.

#### Declaration

**property** AudioFormat: TAudioFormat **read** GetAudioFormat **write** SetAudioFormat **default** DEF\_AudioFormat;

\_\_property TAudioFormat AudioFormat=read=GetAudioFormat, write=SetAudioFormat, **default=0**

Property AudioFormat As TxAudioFormat



## Description

Used to select an audio format in the [AudioFormats](#) list.

## See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

---

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

---

## AudioFormats

### TVideoGrabber.AudioFormats

[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

List of audio formats.

## Declaration

**property** AudioFormats: **string read** GetAudioFormats;

\_\_property wchar\_t \*AudioFormats=read=GetAudioFormats

Property AudioFormats As String

## Description

List of audio formats that can be used with the [current audio capture device](#) .

## See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

---

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

---

## AudioInput

### TVideoGrabber.AudioInput

[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects the current audio input.

## Declaration

**property** AudioInput: LongInt **read** GetAudioInput **write** SetAudioInput;

\_\_property int AudioInput=read=GetAudioInput, write=SetAudioInput, **nodefault**

Property AudioInput As Long

## Description

Used to select/retrieve the current audio input in the [AudioInputs](#) list.



**See Also**

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

---

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

---

**AudioInputBalance****TVideoGrabber.AudioInputBalance**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets/retrieves the audio input balance.

**Declaration**

**property** AudioInputBalance: LongInt **read** GetAudioInputBalance **write** SetAudioInputBalance;

\_\_property **int** AudioInputBalance=read=GetAudioInputBalance, write=SetAudioInputBalance, **nodefault**

Property AudioInputBalance As Long

**Description**

Used to set/retrieve the audio input balance of the current [audio capture device](#).

The valid range is -32768...32767.

0 is the center point.

---

Created with the Standard Edition of HelpNDoc: [Easily Add Encryption and Password Protection to Your PDFs](#)

---

**AudioInputLevel****TVideoGrabber.AudioInputLevel**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets/retrieves the audio input level.

**Declaration**

**property** AudioInputLevel: LongInt **read** GetAudioInputLevel **write** SetAudioInputLevel;

\_\_property **int** AudioInputLevel=read=GetAudioInputLevel, write=SetAudioInputLevel, **nodefault**

Property AudioInputLevel As Long

**Description**

Used to set/retrieve the audio input level of the current [audio capture device](#).

The valid range is from 0 to 65535.

0 indicates that the recording level is off; the value 65535 indicates that the recording level is at full volume.

**See Also**

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

## AudioInputMono

### TVideoGrabber.AudioInputMono

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Disables the stereo mode of the audio input.

#### Declaration

**property** AudioInputMono: Boolean **read** GetAudioInputMono **write** SetAudioInputMono;

\_\_property **bool** AudioInputMono=read=GetAudioInputMono, write=SetAudioInputMono, **nodefault**

Property AudioInputMono As Boolean

#### Description

Used to switch the audio input in "mono" mode of the current [audio capture device](#).  
When disabled, the audio input is in "stereo" mode.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

## AudioInputs

### TVideoGrabber.AudioInputs

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of audio inputs available.

#### Declaration

**property** AudioInputs: **string** **read** GetAudioInputs;

\_\_property wchar\_t \*AudioInputs=read=GetAudioInputs

Property AudioInputs As String

#### Description

Used to retrieve the list of the audio inputs available on the [current audio capture device](#) .  
This list is updated when the [OnAudioDeviceSelected](#) event occurs (when an audio capture device is selected with [AudioDevice](#) ).  
This list can be assigned to list based controls. E.g.:  
*ComboBox1.Items.Text := VideoGrabber1.AudioInputs;*  
*ComboBox1.ItemIndex := VideoGrabber1.AudioInput;*

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#)

[IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#)  
[SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## AudioInputsCount

### TVideoGrabber.AudioInputsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of audio inputs.

#### Declaration

**property** AudioInputsCount: LongInt **read** GetAudioInputsCount;

\_\_property **int** AudioInputsCount=read=GetAudioInputsCount, **nodefault**

Property AudioInputsCount As Long

#### Description

Number of audio inputs in the [AudioInputs](#) list.

The AudioInput index (used to select the current audio input) is in the **0 .. AudioInputsCount - 1** range.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#)  
[AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#)  
[AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioSource](#)  
[IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#)  
[IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#)  
[SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## AudioPeakEvent

### TVideoGrabber.AudioPeakEvent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Activates the [OnAudioPeak](#) event.

#### Declaration

**property** AudioPeakEvent: Boolean **read** GetAudioPeakEvent **write** SetAudioPeakEvent;

\_\_property **bool** AudioPeakEvent=read=GetAudioPeakEvent, write=SetAudioPeakEvent, **default false**

Property AudioPeakEvent As Boolean

#### Description

Used to activate the [OnAudioPeak](#) event, that returns PCM audio peaks.

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## AudioRecording

### TVideoGrabber.AudioRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

**ber**

Enables/disables audio stream capture during [recording](#) .

**Declaration**

**property** AudioRecording: Boolean **read** GetAudioRecording **write** SetAudioRecording;

\_\_property **bool** AudioRecording=read=GetAudioRecording, write=SetAudioRecording, **nodefault**

Property AudioRecording As Boolean

**Description**

Used to enable/disable audio stream capture during [recording](#) .

If disabled, only the video stream is be captured.

**See Also**

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Easily share your documentation with the world through a beautiful website

**AudioRenderer****TVideoGrabber.AudioRenderer**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects the current audio renderer.

**Declaration**

**property** AudioRenderer: LongInt **read** GetAudioRenderer **write** SetAudioRenderer **default** DEF\_AudioRenderer;

\_\_property **int** AudioRenderer==GetAudioRenderer, write=SetAudioRenderer, **default=-1**;

**Description**

Used to select the current audio capture device in the [AudioRenderers](#) list.

The default value -1 uses the default renderer.

It is possible to select the audio renderer programmatically by its name (as it appears in the [AudioRenderers](#) list) by using the [FindIndexInListByName](#) function, e.g.:

```
procedure TfrmMainForm.Button1Click(Sender: TObject);
var
  i: LongInt;
begin
  i := VideoGrabber.FindIndexInListByName (VideoGrabber.AudioRenderers, 'AC97', true,
```

```

    if i > -1 then begin
        VideoGrabber.AudioRenderers := i;
    end;
end;
end;

```

---

Created with the Standard Edition of HelpNDoc: [Qt Help documentation made easy](#)

---

## AudioRendererName

### TVideoGrabber.AudioRendererName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Name of the current audio renderer.

#### Declaration

**property** AudioRendererName: **string read** GetAudioRendererName;

**\_\_property** wchar\_t \*AudioRendererName==GetAudioRendererName;

#### Description

Used to retrieve the name of the audio capture device selected by [AudioRenderer](#) in the [AudioRenderers](#) list.

---

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with a Help Authoring Tool](#)

---

## AudioRenderers

### TVideoGrabber.AudioRenderers

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the audio renderers available on the current platform.

#### Declaration

**property** AudioRenderers: **string read** GetGlobal\_AudioRenderers;

**\_\_property** wchar\_t \*AudioRenderers==GetGlobal\_AudioRenderers;

#### Description

Used to retrieve a string that contains the list of the audio renderers available on the current platform.

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := AudioRenderers;*

*ComboBox1.ItemIndex := VideoGrabber1.AudioRenderer; (\* index in the AudioRenderers list. \*)*

It is possible to retrieve programmatically the index of an audio renderer by using the [FindIndexInListByName](#) function as follows:

*VideoGrabber.AudioRenderer := VideoGrabber.FindIndexInListByName (VideoGrabber.AudioRenderers, AudioRendererName);*

**Note:** in Delphi and C++Builder versions of the component, this property is available also as a [AudioRenderers](#) global variable.

---

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

---

## AudioRenderersCount

### TVideoGrabber.AudioRenderersCount

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Number of audio renderers available on the current platform

#### Declaration

**property** AudioRenderersCount: LongInt **read** GetGlobal\_AudioRenderersCount;

**\_\_property** int AudioRenderersCount==GetGlobal\_AudioRenderersCount, **nodefault**};

#### Description

Number of audio renderers in the [AudioRenderers](#) list.

**Note:** in Delphi and C++Builder versions of the component, this property is available also as a [AudioRenderersCount](#) global variable.

Created with the Standard Edition of HelpNDoc: [Transform Your Word Document into a Professional eBook with HelpNDoc](#)

## AudioSource

### TVideoGrabber.AudioSource

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Source used for the audio recording.

#### Declaration

**property** AudioSource: TAudioSource **read** GetAudioSource **write** SetAudioSource **default** DEF\_AudioSource;

**\_\_property** TAudioSource AudioSource=read=GetAudioSource, write=SetAudioSource, **default**=0

Property AudioSource as TxAudioSource

#### Description

Specifies the source used for the audio recording:

#### as Default:

uses the audio output of the [video capture device](#) if any, otherwise uses the current [audio capture device](#).

always uses the current [audio capture device](#) for the audio recording, even if the video capture device exposes an audio output.

#### as UseExternalAudio:

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## AudioStreamNumber

### TVideoGrabber.AudioStreamNumber

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the audio stream to use

#### Declaration

**property** AudioStreamNumber: LongInt **read** GetAudioStreamNumber **write** SetAudioStreamNumber **default** -1;

**\_\_property** int AudioStreamNumber==GetAudioStreamNumber, write=SetAudioStreamNumber, **default**--1;

property AudioStreamNumber as Long

#### Description

If the clip contains more than one audio stream, use this property to select the audio stream to render

E.g.:

AudioStreamNumber = -1 (default) -> all the audio streams are rendered

AudioStreamNumber = 0 -> only the 1st audio stream is rendered

AudioStreamNumber = 1 -> only the 2nd audio stream is rendered

...

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## AudioSyncAdjustment

### TVideoGrabber.AudioSyncAdjustment

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Introduces an audio delay.

#### Declaration

**property** AudioSyncAdjustment: LongInt **read** GetAudioSyncAdjustment **write** SetAudioSyncAdjustment **default** DEF\_AudioSyncAdjustment;

**\_\_property** int AudioSyncAdjustment==GetAudioSyncAdjustment, write=SetAudioSyncAdjustment, **default**=0;

property AudioSyncAdjustment as Long



## Description

Used to introduce an audio delay, when the video and audio are not synchronized.  
Useful mainly when the audio and video streams do not come out from the same capture device.  
This property specifies a number of audio samples that are delayed, in order to get the audio and video in sync.

To use this property:

- enable [AudioSyncAdjustmentEnabled](#)
- set AudioSyncAdjustment with a number of audio samples to delay, greater than 0

You can make a quick test of this feature with MainDemo.exe -> "audio tab"

## See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Easy EBook and documentation generator](#)

## AudioSyncAdjustmentEnabled

### TVideoGrabber.AudioSyncAdjustmentEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the audio sync adjustment.

## Declaration

**property** AudioSyncAdjustmentEnabled: Boolean **read** GetAudioSyncAdjustmentEnabled **write** SetAudioSyncAdjustmentEnabled **default** DEF\_AudioSyncAdjustmentEnabled;

**\_\_property bool** AudioSyncAdjustmentEnabled==GetAudioSyncAdjustmentEnabled, write=SetAudioSyncAdjustmentEnabled, **default**=0;

property AudioSyncAdjustmentEnabled as Boolean

## Description

Used to enable the audio sync adjustment.  
See [AudioSyncAdjustment](#).

Created with the Standard Edition of HelpNDoc: [Make the switch to CHM with HelpNDoc's hassle-free WinHelp HLP to CHM conversion tool](#)

## AudioVolume

### TVideoGrabber.AudioVolume

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Audio volume of the current [audio capture device](#) .



## Declaration

**property** AudioVolume: LongInt **read** GetAudioVolume **write** SetAudioVolume;

\_\_property int AudioVolume=read=GetAudioVolume, write=SetAudioVolume, **nodefault**

Property AudioVolume As Long

## Description

Used to set/retrieve the audio volume of the current [audio capture device](#) when [AudioDeviceRendering](#) is enabled.

The valid range is from 0 (min level) to 65535 (max leve).

**This a rendering audio property only, it does not apply to the audio recording.**

## See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [MuteAudioRendering](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

## AutoFileName

### TVideoGrabber.AutoFileName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Format of file names generated automatically.

## Declaration

**property** AutoFileName: TAutoFileName **read** GetAutoFileName **write** SetAutoFileName **default** DEF\_AutoFileName;

\_\_property TAutoFileName AutoFileName=read=GetAutoFileName, write=SetAutoFileName, **default=0**

Property AutoFileName As TxAutoFileName

## Description

Sets or retrieves the format (sequential or date/time) used to generate file names automatically during [frame capture](#) or [recording](#) .

## See Also

[AutoFileNameDateTimeFormat](#) [AutoFilePrefix](#) [CaptureFileExt](#) [Last\\_BurstFrameCapture\\_FileName](#) [Last\\_CaptureFrameTo\\_FileName](#) [RecordingFileName](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## AutoFileNameDateTimeFormat

### TVideoGrabber.AutoFileNameDateTimeFormat

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specify the date/time format for recording file name

## Declaration

**property** AutoFileNameDateTimeFormat: **string** **read** GetAutoFileNameDateTimeFormat **write** SetAutoFileNameDateTimeFormat;

**\_\_property** wchar\_t \*AutoFileNameDateTimeFormat=read=GetAutoFileNameDateTimeFormat, write=SetAutoFileNameDateTimeFormat

Property AutoFileNameDateTimeFormat As String

### Description

Lets customize the date/time format when the recording file names are generated automatically and [AutoFileName](#) is set to **fn\_DateTime**

By default the date/time format of the file names is:

**yymmdd\_hhmmss\_zzz**

By default [AutoFilePrefix](#) is set to "vg" (default), so the recording file name is e.g.

**vg130213\_151847\_352.avi**

By example if you customize it as follows:

VideoGrabber.AutoFileName = fn\_DateTime;

VideoGrabber.AutoFileNameDateTimeFormat = "dd-mm-yy\_hh-mm-ss"

VideoGrabber.AutoFilePrefix = "file\_"

the recording file name will be **file\_13-02-13\_15-27-45.avi**

### See Also

[AutoFileName](#) [AutoFilePrefix](#) [CaptureFileExt](#) [Last\\_BurstFrameCapture\\_FileName](#)  
[Last\\_CaptureFrameTo\\_FileName](#) [RecordingFileName](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## AutoFileNameMinDigits

### TVideoGrabber.AutoFileNameMinDigits

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the number of 0 to pad auto-generated file names.

### Declaration

**property** AutoFileNameMinDigits: **LongInt** read GetAutoFileNameMinDigits **write** SetAutoFileNameMinDigits **default** DEF\_AutoFileNameMinDigits;

**\_\_property** **int** AutoFileNameMinDigits=read=GetAutoFileNameMinDigits, write=SetAutoFileNameMinDigits, **default**=6

Property AutoFileNameMinDigits As Long

### Description

Used to specify the number of padding "0" used to generate a file name when [AutoFileName](#) = fn\_Sequential.

E.g.:

- if AutoFileNameMinDigits = 6, the file name will be generated e.g. as

vg000001.avi

vg000002.avi

vg000003.avi

...

- if AutoFileNameMinDigits = 3, the file name will be generated e.g. as

vg001.avi

vg002.avi

vg003.avi

...

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## AutoFilePrefix

### TVideoGrabber.AutoFilePrefix

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Prefix of file names generated automatically.

#### Declaration

**property** AutoFilePrefix: **string** **read** GetAutoFilePrefix **write** SetAutoFilePrefix;

\_\_property wchar\_t \*AutoFilePrefix=read=GetAutoFilePrefix, write=SetAutoFilePrefix

Property AutoFilePrefix As String

#### Description

Sets or retrieves the prefix added at the beginning of file names generated automatically during [frame capture](#) or [recording](#) .

#### See Also

[AutoFileName](#) [AutoFileNameDateTimeFormat](#) [CaptureFileExt](#) [Last\\_BurstFrameCapture\\_FileName](#) [Last\\_CaptureFrameTo\\_FileName](#) [RecordingFileName](#)

Created with the Standard Edition of HelpNDoc: [Make the switch to CHM with HelpNDoc's hassle-free WinHelp HLP to CHM conversion tool](#)

## AutoRefreshPreview

### TVideoGrabber.AutoRefreshPreview

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

If enabled, preview is automatically restarted each time a property that requires to restart preview is modified.

#### Declaration

**property** AutoRefreshPreview: Boolean **read** GetAutoRefreshPreview **write** SetAutoRefreshPreview **default** DEF\_AutoRefreshPreview;

\_\_property **bool** AutoRefreshPreview=read=GetAutoRefreshPreview, write=SetAutoRefreshPreview,

**default=1**

Property AutoRefreshPreview As Boolean

### Description

If enabled, preview is automatically restarted each time a property that requires the preview to be restarted (to reflect the new value) is modified.

### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## AutoStartPlayer

### TVideoGrabber.AutoStartPlayer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to specify whether a clip must start playback automatically when [OpenPlayer](#) is called.

### Declaration

**property** AutoStartPlayer: Boolean **read** GetAutoStartPlayer **write** SetAutoStartPlayer **default** DEF\_AutoStartPlayer;

\_\_property **bool** AutoStartPlayer=read=GetAutoStartPlayer, write=SetAutoStartPlayer, **default=1**

Property AutoStartPlayer As Boolean

### Description

If enabled, clips start playing automatically when they are opened.

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

## AVIDurationUpdated

### TVideoGrabber.AVIDurationUpdated

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Activates the [OnAVIDurationUpdated](#) event.

### Declaration

**property** AVIDurationUpdated: Boolean **read** GetAVIDurationUpdated **write** SetAVIDurationUpdated;

\_\_property **bool** AVIDurationUpdated=read=GetAVIDurationUpdated, write=SetAVIDurationUpdated,  
**default false**

Property AVIDurationUpdated As Boolean

#### Description

When activated and the AVI recording ends:

- TVideoGrabber fixes a possible wrong AVI duration in the AVI file format, that can be caused e.g. by a large amount of dropped frames.
- the [OnAVIDurationUpdated](#) event occurs, that gives more control on the duration and frame rate interval.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

## AVIFormatOpenDML

### TVideoGrabber.AVIFormatOpenDML

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the index format of the AVI file (AVI 1.0 or AVI 2.0 OpenDML).

#### Declaration

**property** AVIFormatOpenDML: Boolean **read** GetAVIFormatOpenDML **write** SetAVIFormatOpenDML  
**default true**;

\_\_property **bool** AVIFormatOpenDML=read=GetAVIFormatOpenDML, write=SetAVIFormatOpenDML,  
**default=1**

Property AVIFormatOpenDML As Boolean

#### Description

Used to specify the index format of the AVI file created during [recording](#) .

**false**: indicates an AVI 1.0 compatible index format.

**true**: indicates an AVI 2.0 index format.

AVI 2.0 index format allows for increased AVI file size (greater than 1 GB), hierarchical indexing, incremental growth of files, and minimal disk seeks.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#)

[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## AVIFormatOpenDMLCompatibilityIndex

### TVideoGrabber.AVIFormatOpenDMLCompatibilityIndex

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Specifies if the AVI 2.0 OpenDML file includes an AVI 1.0 index.

#### Declaration

**property** AVIFormatOpenDMLCompatibilityIndex: Boolean **read** GetAVIFormatOpenDMLCompatibilityIndex **write** SetAVIFormatOpenDMLCompatibilityIndex **default** true;

\_\_property bool AVIFormatOpenDMLCompatibilityIndex=read=GetAVIFormatOpenDMLCompatibilityIndex, write=SetAVIFormatOpenDMLCompatibilityIndex, **default**=1

Property AVIFormatOpenDMLCompatibilityIndex As Boolean

#### Description

When [AVIFormatOpenDML](#) is enabled:

**true** : specifies that both an AVI 2.0 index format **AND** an AVI 1.0 index is included in the AVI 2.0 format (for Video for Windows compatibility, allows to get information about the video clip by displaying the file properties)

**false**: specifies that only an AVI 2.0 index is included.

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## BackgroundColor

### TVideoGrabber.BackgroundColor

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Color of the background when the video size is smaller than the video window.

#### Declaration

**property** BackgroundColor: TColor **read** GetBackgroundColor **write** SetBackgroundColor **default** DEF\_BackgroundColor;

\_\_property Graphics::TColor BackgroundColor=read=GetBackgroundColor, write=SetBackgroundColor, **default**=0

Property BackgroundColor As OLE\_COLOR

#### Description

Color of the background when the video size is smaller than the video window.

Used when the video window is not embedded and KeepAspectRatio is enabled.

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

## BorderStyle

### TVideoGrabber.BorderStyle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enable or disable border.

#### Declaration

**property** BorderStyle: TBorderStyle **read** GetBorderStyle **write** SetBorderStyle **default** DEF\_BorderStyle;

\_\_property Forms::TBorderStyle BorderStyle=read=GetBorderStyle, write=SetBorderStyle, **default**=1

Property BorderStyle As TxBorderStyle

#### Description

Used to enable or disable the control border.

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## BurstCount

### TVideoGrabber.BurstCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of frames to capture.

#### Declaration

**property** BurstCount: LongInt **read** GetBurstCount **write** SetBurstCount **default** DEF\_BurstCount;

\_\_property **int** BurstCount=read=GetBurstCount, write=SetBurstCount, **default**=3

Property BurstCount As Long

#### Description

Used to set or retrieve the number of frames to capture when [BurstMode](#) will be enabled.  
A "0" value means "don't stop capturing frames".

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)



## BurstInterval

### TVideoGrabber.BurstInterval

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Number of frames to skip between 2 captured frames.

#### Declaration

**property** BurstInterval: LongInt **read** GetBurstInterval **write** SetBurstInterval **default** DEF\_BurstInterval;

\_\_property **int** BurstInterval=read=GetBurstInterval, write=SetBurstInterval, **default**=3

Property BurstInterval As Long

#### Description

Sets or retrieves the number of frames to skip between 2 captured frames.

A "0" value captures all the frames (no frame skipped).

*E.g.: BurstInterval = 20 means 1 frame captured every 20 frames.*

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last\\_BurstFrameCapture](#) [FileName](#) [Last\\_CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

## BurstMode

### TVideoGrabber.BurstMode

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Enables/disables the automated frame capture process.

#### Declaration

**property** BurstMode: Boolean **read** GetBurstMode **write** SetBurstMode **default** DEF\_BurstMode;

\_\_property **bool** BurstMode=read=GetBurstMode, write=SetBurstMode, **default**=0

Property BurstMode As Boolean

#### Description

Used to enable/disable the automated frame capture process.

The number of frames captured depends of the [BurstCount](#) and [BurstInterval](#) parameters.

The destination of the captured frame depends of the [BurstType](#) parameter.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last\\_BurstFrameCapture](#) [FileName](#) [Last\\_CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#)



## [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

## BurstType

### TVideoGrabber.BurstType

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Destination of captured frames.

#### Declaration

**property** BurstType: TFrameCaptureDest **read** GetBurstType **write** SetBurstType **default** DEF\_BurstType;

\_\_property TFrameCaptureDest BurstType=read=GetBurstType, write=SetBurstType, **default**=0

Property BurstType As TFrameCaptureDest

#### Description

Used to set or retrieve the destination of captured frames:

- **fc\_TBitmap**: to memory bitmap,
- **fc\_BMPFile**: to a BMP file,
- **fc\_JPEGFile**: to a JPEG file,
- **fc\_Clipboard**: to the clipboard (CF\_BITMAP format)

The type of this property is [TFrameCaptureDest](#).

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## Busy

### TVideoGrabber.Busy

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if TVideoGrabber is in an intermediary state.

#### Declaration

**property** Busy: Boolean **read** GetBusy;

\_\_property **bool** Busy=read=GetBusy, **nodefault**

Property Busy As Boolean

#### Description

Returns true if TVideoGrabber is in an intermediary state (e.g. during preview or recording startup) and cannot execute commands immediately.

However, commands and property changes received by the component during a busy state are queued and executed as soon as the component becomes available.

## BusyCursor

### TVideoGrabber.BusyCursor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Cursor displayed when TVideoGrabber is in a [busy](#) state.

#### Declaration

**property** BusyCursor: TCursor **read** GetBusyCursor **write** SetBusyCursor **default** DEF\_BusyCursor;

\_\_property Controls::TCursor BusyCursor=read=GetBusyCursor, write=SetBusyCursor, **default=- 11**

Property BusyCursor As TxCursors

#### Description

Sets or retrieves the cursor displayed while the component processes significant tasks (like reencoding or copying captured data, restarting preview or capture).

The BusyCursor is displayed when the task begins, and then, the [NormalCursor](#) is displayed when the task ends.

Set BusyCursor to **crDefault** to disable this feature. In this case NormalCursor is ignored.

## CameraControlSettings

### TVideoGrabber.CameraControlSettings

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to enable or disable the camera control settings.

#### Declaration

**property** CameraControlSettings: Boolean **read** GetCameraControlSettings **write** SetCameraControlSettings **default** DEF\_CameraControlSettings;

\_\_property **bool** CameraControlSettings=read=GetCameraControlSettings, write=SetCameraControlSettings, **default=1**

Property CameraControlSettings As Boolean

#### Description

Enables or disables the automatic saving and restoring of [camera control](#) settings.

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

## CaptureFileExt

### TVideoGrabber.CaptureFileExt

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Extension of AVI files names generated automatically.

### Declaration

**property** CaptureFileExt: **string read** GetCaptureFileExt **write** SetCaptureFileExt;

\_\_property wchar\_t \*CaptureFileExt=read=GetCaptureFileExt, write=SetCaptureFileExt

Property CaptureFileExt As String

### Description

Default extension used to create video clips when the file name is generated [automatically](#) .

Leave it blank if you want TVideoGrabber to generate it automatically. In this case it assigns:

- "avi" for standard AVI files,
- "mpg" for MPEG files.

### See Also

[AutoFileName](#) [AutoFileNameDateTimeFormat](#) [AutoFilePrefix](#) [Last\\_BurstFrameCapture\\_FileName](#) [Last\\_CaptureFrameTo\\_FileName](#) [RecordingFileName](#)

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

## ColorKey

### [TVideoGrabber](#).ColorKey

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Used to modify the default color key used for window transparency.

### Declaration

**property** ColorKey: TColorKey **read** GetColorKey **write** SetColorKey **default** DEF\_ColorKey;

\_\_property bool ColorKey=read=GetColorKey, write=SetColorKey, **default**=0

Property ColorKey As OLECOLOR

### Description

Used to change the default color keys used for window transparency and related effects.

Enable the window transparency with [ColorKeyEnabled](#), then invoke [SetWindowTransparency](#) to set your form's or window transparency.

*You can find sample code in the transparency checkbox code of the "display" tab of the MainDemo project.*

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## ColorKeyEnabled

### [TVideoGrabber](#).ColorKeyEnabled

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Enables color-key based effects, like windows transparency.

**Declaration**

**property** ColorKeyEnabled: Boolean **read** GetColorKeyEnabled **write** SetColorKeyEnabled **default** DEF\_ColorKeyEnabled;

\_\_**property bool** ColorKeyEnabled=read=GetColorKeyEnabled, write=SetColorKeyEnabled, **default**=0

Property ColorKeyEnabled As Boolean

**Description**

Used to enable color-key based effects, like window transparency.

The default color keys can be changed with the [ColorKey](#) property.

Then, when the preview, recording or play back starts, the [OnColorKeyChange](#) event returns the effective color key to use.

See the *VideoGrabber1ColorKeyChange* event of the *MainDemo* demo project for sample code.

**See Also**

[TOnColorKeyChange](#) [OnColorKeyChange](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

**CompressionMode****TVideoGrabber.CompressionMode**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Compression mode applied to recording.

**Declaration**

**property** CompressionMode: TCompressionMode **read** GetCompressionMode **write** SetCompressionMode **default** DEF\_CompressionMode;

\_\_**property TCompressionMode** CompressionMode=read=GetCompressionMode, write=SetCompressionMode, **default**=0

Property CompressionMode As TxCompressionMode

**Description**

Used to set/retrieve the [TCompressionMode](#) compression mode applied to recording.

**See Also**

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

**CompressionType****TVideoGrabber.CompressionType**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Compression type applied to recording.

**Declaration**

**property** CompressionType: TCompressionType **read** GetCompressionType **write** SetCompressionType  
**default** DEF\_CompressionType;

\_\_property TCompressionType CompressionType=read=GetCompressionType,  
 write=SetCompressionType, **default**=0

Property CompressionType As TxCompressionType

**Description**

Used to set/retrieve the [TCompressionType](#) compression type applied to recording.

**See Also**

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#)  
[AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#)  
[CompressionMode](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#)  
[OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#)  
[SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#)  
[VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#)  
[VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#)  
[VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

**Cropping\_Enabled****TVideoGrabber.Cropping\_Enabled**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the video cropping

**Declaration**

**property** Cropping\_Enabled: Boolean **read** GetCropping\_Enabled **write** SetCropping\_Enabled **default**  
 DEF\_Cropping\_Enabled;

\_\_property **bool** Cropping\_Enabled=read=GetCropping\_Enabled, write=SetCropping\_Enabled, **default**=0

Property Cropping\_Enabled As Boolean

**Description**

Used to enable/disable the cropping of the video stream.

**See Also**

[Cropping and zooming](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#)  
[Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

Created with the Standard Edition of HelpNDoc: [Easily share your documentation with the world through a beautiful website](#)

**Cropping\_Height****TVideoGrabber.Cropping\_Height**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Height of the cropping area.

**Declaration**

**property** Cropping\_Height: LongInt **read** GetCropping\_Height **write** SetCropping\_Height **default**

DEF\_Cropping\_Height;

\_\_property **int** Cropping\_Height=read=GetCropping\_Height, write=SetCropping\_Height, **default**=120

Property Cropping\_Height As Long

#### Description

Specifies the height of the cropping area. The value must be lower or equal to the [VideoHeight](#) property.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

## Cropping\_Outbounds

### TVideoGrabber.Cropping\_Outbounds

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Keeps the cropping area centered when zooming near the edges.

#### Declaration

**property** Cropping\_Outbounds: Boolean **read** GetCropping\_Outbounds **write** SetCropping\_Outbounds **default** DEF\_Cropping\_Outbounds;

\_\_property **bool** Cropping\_Outbounds=read=GetCropping\_Outbounds, write=SetCropping\_Outbounds, **default**=0

Property Cropping\_Outbounds As Boolean

#### Description

Useful when zooming near the edges of the video window.

When this property is enabled, moving the [Cropping\\_X](#) and [Cropping\\_Y](#) values against the edge of the video window makes the edge displayed at the center of the control (with a background border), allowing to zoom near the edge while keeping the image centered.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

## Cropping\_Width

### TVideoGrabber.Cropping\_Width

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Width of the cropping area.

#### Declaration

**property** Cropping\_Width: LongInt **read** GetCropping\_Width **write** SetCropping\_Width **default** DEF\_Cropping\_Width;

\_\_property **int** Cropping\_Width=read=GetCropping\_Width, write=SetCropping\_Width, **default**=160

Property Cropping\_Width As Long

#### Description

Specifies the width of the cropping area. The value must be lower or equal to the [VideoWidth](#) property.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

---

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

---

## Cropping\_X

### TVideoGrabber.Cropping\_X

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

X axis location of the cropping area.

#### Declaration

**property** Cropping\_X: LongInt **read** GetCropping\_X **write** SetCropping\_X **default** DEF\_Cropping\_X;

\_\_property **int** Cropping\_X=read=GetCropping\_X, write=SetCropping\_X, **default**=0

Property Cropping\_X As Long

#### Description

Used to set/retrieve the X axis location of the cropping area.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

---

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

---

## Cropping\_XMax

### TVideoGrabber.Cropping\_XMax

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Maximum [Cropping\\_X](#) position allowed, depending of the current video size.

#### Declaration

**property** Cropping\_XMax: LongInt **read** GetCropping\_XMax;

\_\_property **int** Cropping\_XMax=read=GetCropping\_XMax, **nodefault**

Property Cropping\_XMax As Long

#### Description

Returns the maximum [Cropping\\_X](#) position allowed, depending of the video size.  
This property is refreshed when the [OnResizeVideo](#) event occurs.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_Y](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

---

## Cropping\_Y

### TVideoGrabber.Cropping\_Y

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Y axis location of the cropping area.

#### Declaration

**property** Cropping\_Y: LongInt **read** GetCropping\_Y **write** SetCropping\_Y **default** DEF\_Cropping\_Y;  
 \_\_property **int** Cropping\_Y=read=GetCropping\_Y, write=SetCropping\_Y, **default**=0

Property Cropping\_Y As Long

#### Description

Used to set/retrieve the Y axis location of the cropping area.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_YMax](#) [Cropping\\_Zoom](#)

Created with the Standard Edition of HelpNDoc: [Protect Your Confidential PDFs with These Simple Security Measures](#)

## Cropping\_YMax

### TVideoGrabber.Cropping\_YMax

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Maximum [Cropping\\_Y](#) position allowed, depending of the current video size.

#### Declaration

**property** Cropping\_YMax: LongInt **read** GetCropping\_YMax;  
 \_\_property **int** Cropping\_YMax=read=GetCropping\_YMax, **nodefault**

Property Cropping\_YMax As Long

#### Description

Returns the maximum [Cropping\\_Y](#) position allowed, depending of the video size.  
 This property is refreshed when the [OnResizeVideo](#) event occurs.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_Zoom](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## Cropping\_Zoom

### TVideoGrabber.Cropping\_Zoom

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Zooming coefficient of the cropping area displayed.

#### Declaration



**property** Cropping\_Zoom: Double **read** GetCropping\_Zoom **write** SetCropping\_Zoom;  
 \_\_property **double** Cropping\_Zoom=read=GetCropping\_Zoom, write=SetCropping\_Zoom

Property Cropping\_Zoom As Double

#### Description

Sets/retrieves the zooming coefficient of the cropping area displayed.

#### See Also

[Cropping and zooming](#) [Cropping\\_Enabled](#) [Cropping\\_Height](#) [Cropping\\_Outbounds](#) [Cropping\\_Width](#) [Cropping\\_X](#) [Cropping\\_XMax](#) [Cropping\\_Y](#) [Cropping\\_YMax](#)

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## CurrentFrameRate

### TVideoGrabber.CurrentFrameRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Effective frame rate.

#### Declaration

**property** CurrentFrameRate: Double **read** GetCurrentFrameRate;  
 \_\_property **double** CurrentFrameRate=read=GetCurrentFrameRate

Property CurrentFrameRate As Double

#### Description

Retrieves the effective frame rate, that can differ from the [FrameRate](#) that has been set. Useful to know when the FrameRate property cannot be set, e.g. with video capture devices that have a fixed frame rate depending of the video format (e.g. like DV sources). Note that some video capture devices accept any stupid value and return this stupid value even if they limit this value to an acceptable value.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## CurrentState

### TVideoGrabber.CurrentState

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the [TCurrentState](#) current state of TVideoGrabber.

#### Declaration

**property** CurrentState: TCurrentState **read** GetCurrentState;  
 \_\_property TCurrentState CurrentState=read=GetCurrentState, **nodefault**

Property CurrentState As TxCurrentState

## Description

Used to retrieve the [TCurrentState](#) current state of TVideoGrabber (down, previewing, capturing to AVI, reencodinging or playing back).

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

## DeliveredFrames

### TVideoGrabber.DeliveredFrames

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Current number of delivered frames.

## Declaration

**property** DeliveredFrames: LargeInteger **read** GetDeliveredFrames;

\_\_property \_\_int64 DeliveredFrames=read=GetDeliveredFrames

Property DeliveredFrames As Double

## Description

Used to retrieve the current number of delivered frames since [preview](#) or [recording](#) started.

## See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## DirectShowFilters

### TVideoGrabber.DirectShowFilters

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the DirectShow filters available

## Declaration

**function** DirectShowFilters: **string**;

\_\_property wchar\_t \* \_\_fastcall DirectShowFilters();

property DirectShowFilters as string

## Description

Returns the list of the DirectShow filters available on the current platform.

Created with the Standard Edition of HelpNDoc: [How to Protect Your PDFs with Encryption and Passwords](#)

## DirectShowFiltersCount

### TVideoGrabber.DirectShowFiltersCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

## ber

Number of DirectShow filters available

### Declaration

property DirectShowFiltersCount: LongInt read GetGlobal\_DirectShowFiltersCount;

\_\_property int DirectShowFiltersCount=read=GetGlobal\_DirectShowFiltersCount, nodefaut

Function DirectShowFiltersCount as Long

### Description

Returns the number of DirectShow filters listed in the [DirectShowFilters](#) list.

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## Display\_Active

### TVideoGrabber.Display\_Active

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the video window.

### Declaration

**property** Display\_Active: Boolean **index** vwActive **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_Active;

\_\_property **bool** Display\_Active=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=0, default=1

Property Display\_Active As Boolean

### Description

Used to enable/disable the video window.

### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## Display\_AlphaBlendEnabled

### TVideoGrabber.Display\_AlphaBlendEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

**ber**

Activates the alpha blending of the video window

**Declaration**

**property** Display\_AlphaBlendEnabled: Boolean **index** vwAlphaBlendEnabled **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_AlphaBlendEnabled;

**\_\_property** bool Display\_AlphaBlendEnabled=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=0, **default**=1

Property Display\_AlphaBlendEnabled As Boolean

**Description**

Used to activate the alpha blending of the current video window with another video window when it is detached from the control (when [Display\\_Embedded](#) = false)

**See Also**

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc

**Display\_AlphaBlendValue****TVideoGrabber.Display\_AlphaBlendValue**
[Prev](#)
[Next](#)
**TVideoGrabber Properties**

Value of the alpha blending of the video window

**Declaration**

**property** Display\_AlphaBlendValue: LongInt **index** vwAlphaBlendValue **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_AlphaBlendValue;

**\_\_property** int Display\_AlphaBlendValue=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=12, **default**=240

Property Display\_AlphaBlendValue As Long

**Description**

Value of the alpha blending of the video window, in the 0..255 range (e.g. 128 = semi-transparent)  
This feature can be used when [Display\\_AlphaBlendEnabled](#) = true and [Display\\_Embedded](#) = false

**See Also**

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor](#)

[Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#)  
[Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#)  
[Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#)  
[DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#)  
[DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#)  
[DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#)  
[DualDisplay\\_SetLocation](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#)  
[DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#)  
[DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#)  
[IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#)  
[SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#)  
[VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [How to Protect Your PDFs with Encryption and Passwords](#)

## Display\_AspectRatio

### TVideoGrabber.Display\_AspectRatio

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Specifies the aspect ratio to use within the video window.

#### Declaration

**property** Display\_AspectRatio: TAspectRatio **index** 0 **read** GetDisplayAspectRatio **write** SetDisplayAspectRatio **default** ar\_Box;

\_\_property TAspectRatio Display\_AspectRatio=read=GetDisplayAspectRatio, write=SetDisplayAspectRatio, index=0, **default**=0

Property Display\_AspectRatio As TAspectRatio

#### Description

Used when [Display\\_AutoSize](#) = false to specify the aspect ratio method that must be used within the video window.

The possible values are described in the [TAspectRatio](#) type.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#)  
[Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#)  
[Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#)  
[Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#)  
[Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#)  
[Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#)  
[DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#)  
[DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#)  
[DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#)  
[DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#)  
[DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#)  
[IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#)  
[SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#)  
[VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

## Display\_AutoSize

### TVideoGrabber.Display\_AutoSize

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

If enabled, the control is resized automatically according to the current video size.

#### Declaration

**property** Display\_AutoSize: Boolean **index** vwAutoSize **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_AutoSize;

\_\_property **bool** Display\_AutoSize=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=1, **default**=1

Property Display\_AutoSize As Boolean

#### Description

Specifies whether the control must be resized automatically according to the current video size.

**disabled:** the control size depends of the Width and Height properties, and the video window is stretched inside.

**enabled:** the control size is automatically modified according to [VideoSize](#) or [UseNearestVideoSize](#) .

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## Display\_Embedded

### TVideoGrabber.Display\_Embedded

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Used to detach/attach the video window from the TVideoGrabber control.

#### Declaration

**property** Display\_Embedded: Boolean **index** vwEmbedded **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_Embedded;

\_\_property **bool** Display\_Embedded=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=2, **default**=1

Property Display\_Embedded As Boolean

#### Description



Used to detach/attach the video window from the TVideoGrabber control.

- **enabled**: the video window is embedded into the TVideoGrabber control,
- **disabled**: the video window is located on the desktop at the [Display\\_Left](#) and [Display\\_Top](#) positions (on the [Display\\_Monitor](#) monitor if more than 1 monitor is installed).

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: iPhone web sites made easy

## Display\_FullScreen

### TVideoGrabber.Display\_FullScreen

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Displays the preview window in full screen mode.

#### Declaration

**property** Display\_FullScreen: Boolean **index** vwFullScreen **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_FullScreen;

**\_\_property** **bool** Display\_FullScreen=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=3, **default**=0

Property Display\_FullScreen As Boolean

#### Description

If enabled, the preview window is displayed in full screen mode when the preview, recording or play back starts.

To exit from the full screen mode disable this property or press the <ESC> key.

In full screen mode all keystrokes are returned by the [OnKeyPress](#) event.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#)

[VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

**Display\_Height****TVideoGrabber.Display\_Height**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the height of the video window, when it is not embedded in the TVideoGrabber control.

**Declaration**

**property** Display\_Height: LongInt **index** vwHeight **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_Height;

\_\_property **int** Display\_Height=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=12, **default**=240

Property Display\_Height As Long

**Description**

Used to specify the height of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by using [Display\\_SetLocation](#) .

**See Also**

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

**Display\_Left****TVideoGrabber.Display\_Left**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the left position of the video window, when it is not embedded in the TVideoGrabber control.

**Declaration**

**property** Display\_Left: LongInt **index** vwLeft **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_Left;

\_\_property **int** Display\_Left=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=13, **default**=0



Property Display\_Left As Long

### Description

Specifies the left position of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by using [Display\\_SetLocation](#) .

### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Transform Your Help Documentation Process with a Help Authoring Tool

## Display\_Monitor

### TVideoGrabber.Display\_Monitor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the monitor used to display the video window.

### Declaration

**property** Display\_Monitor: LongInt **index** vwMonitor **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_Monitor;

\_\_property **int** Display\_Monitor=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=8, **default**=0

Property Display\_Monitor As Long

### Description

Used to specify the monitor used to display the video window.

The value is in the ( 0..[MonitorsCount](#) -1 ) range.

### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

[VideoWidth PreferredAspectRatio](#)Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)**Display\_MouseMovesWindow****TVideoGrabber.Display\_MouseMovesWindow**[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Specifies whether the mouse moves the video window or returns mouse events for this window.

**Declaration**

**property** Display\_MouseMovesWindow: Boolean **index** vwMouseMovesWindow **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_MouseMovesWindow;

**\_\_property** bool Display\_MouseMovesWindow=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=6, **default**=1

Property Display\_MouseMovesWindow As Boolean

**Description**

When the video window is not embedded in the TVideoGrabber control:

- **if enabled:** the mouse moves the video window,
- **if disabled:** the mouse returns mouse events, e.g. to draw graphic objects over the video window.

**See Also**

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)**Display\_PanScanRatio****TVideoGrabber.Display\_PanScanRatio**[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Adjust the Pan/Scan ratio.

**Declaration**

**property** Display\_PanScanRatio: LongInt **index** vwPanScanRatio **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_PanScanRatio;

**\_\_property** int Display\_PanScanRatio==GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=11, **default**=50;

Property Display\_PanScanRatio as Long

### Description

When [Display\\_AspectRatio](#) = **ar\_PanScan**, this property is used to adjust the Pan/Scan ratio (in the 0..100 range, default value 50)

### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Revolutionize Your CHM Help File Output with HelpNDoc

## Display\_StayOnTop

### TVideoGrabber.Display\_StayOnTop

[Prev](#)
[Next](#)

[TVideoGrabber](#) **Properties**

Specifies whether the video window must stay over other windows, when not embedded.

### Declaration

**property** Display\_StayOnTop: Boolean **index** vwStayOnTop **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_StayOnTop;

\_\_property **bool** Display\_StayOnTop=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=5, **default**=0

Property Display\_StayOnTop As Boolean

### Description

Specifies whether the video window must stay over other windows, when it is not embedded in the TVideoGrabber control.

### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

## Display\_Top

### TVideoGrabber.Display\_Top

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the top position of the video window, when it is not embedded in the TVideoGrabber control.

#### Declaration

**property** Display\_Top: LongInt **index** vwTop **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_Top;

\_\_property **int** Display\_Top=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=14, **default**=0

Property Display\_Top As Long

#### Description

Specifies the top position of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by using [Display\\_SetLocation](#) .

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

## Display\_TransparentColorEnabled

### TVideoGrabber.Display\_TransparentColorEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Activate the color keying of the current video window

#### Declaration

**property** Display\_TransparentColorEnabled: Boolean **index** vwTransparentColorEnabled **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_TransparentColorEnabled;

\_\_property **bool** Display\_TransparentColorEnabled=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=0, **default**=1

Property Display\_TransparentColorEnabled As Boolean

## Description

Used to activate the color keying of the current video window with another video window when it is detached from the control (when [Display\\_Embedded](#) = false)

## See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Easy EPub and documentation editor

## Display\_TransparentColorValue

### TVideoGrabber.Display\_TransparentColorValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Value of the color keying of the video window

## Declaration

**property** Display\_TransparentColorValue: LongInt **index** vwTransparentColorValue **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_TransparentColorValue;

**\_\_property int** Display\_TransparentColorValue=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=12, **default**=240

Property Display\_TransparentColorValue As Long

## Description

Value of the color keying of the video window, expressed in RGB value.

*This feature can be used when [Display\\_TransparentColorEnabled](#) = true and [Display\\_Embedded](#) = false*

## See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)



## Display\_VideoHeight

### TVideoGrabber.Display\_VideoHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the height of the 1st video window.

#### Declaration

**property** Display\_VideoHeight: LongInt **read** GetDisplay\_VideoHeight;

\_\_property **int** Display\_VideoHeight=read=GetDisplay\_VideoHeight, **nodefault**

Property Display\_VideoHeight As Long

#### Description

Used to retrieve the height of the 1st video window.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

## Display\_VideoPortEnabled

### TVideoGrabber.Display\_VideoPortEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the video port, if available on the current video capture device.

#### Declaration

**property** Display\_VideoPortEnabled: Boolean **index** vwVideoPortEnabled **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_VideoPortEnabled;

\_\_property **bool** Display\_VideoPortEnabled=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=7, **default**=1

Property Display\_VideoPortEnabled As Boolean

#### Description

If a video port is available on the current video capture device ([IsVideoPortAvailable](#) returns true), and this

property is true, the video capture device writes directly to video memory on the preview stream. This provides a smoother display, however it is not possible to draw over video frames on the preview stream, even if the [frame grabber](#) is inserted on the preview stream.

If you need to draw over video frames and get the results visible on the preview window, disable this property.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

## Display\_VideoWidth

### TVideoGrabber.Display\_VideoWidth

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the width of the 1st video window.

#### Declaration

**property** Display\_VideoWidth: LongInt **read** GetDisplay\_VideoWidth;

\_\_property **int** Display\_VideoWidth=read=GetDisplay\_VideoWidth, **nodefault**

Property Display\_VideoWidth As Long

#### Description

Used to retrieve the width of the 1st video window.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

## Display\_VideoWindowHandle

### TVideoGrabber.Display\_VideoWindowHandle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the window handle of the 1st video window.

#### Declaration

**property** Display\_VideoWindowHandle: Hwnd **read** GetDisplay\_VideoWindowHandle;

\_\_property Hwnd Display\_VideoWindowHandle=read=GetDisplay\_VideoWindowHandle, **nodefault**

Property Display\_VideoWindowHandle As Long

#### Description

Used to retrieve the window handle of the 1st video window.  
Returns 0 if the 1st video window does not exist.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

## Display\_Visible

### TVideoGrabber.Display\_Visible

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Shows / hide the video window

#### Declaration

**property** Display\_Visible: Boolean **index** vwVisible **read** GetDisplayBoolProperties **write** SetDisplayBoolProperties **default** DEF\_Display\_Visible;

\_\_property bool Display\_Visible=read=GetDisplayBoolProperties, write=SetDisplayBoolProperties, index=1, **default**=1

Property Display\_Visible As Boolean

#### Description

Used to show / hide the video window.



Enabled by default.

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## Display\_Width

### TVideoGrabber.Display\_Width

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the width of the video window, when it is not embedded in the TVideoGrabber control.

#### Declaration

**property** Display\_Width: LongInt **index** vwWidth **read** GetDisplayLongIntProperties **write** SetDisplayLongIntProperties **default** DEF\_Display\_Width;

\_\_property **int** Display\_Width=read=GetDisplayLongIntProperties, write=SetDisplayLongIntProperties, index=11, **default**=320

Property Display\_Width As Long

#### Description

Used to specify the width of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by using [Display\\_SetLocation](#) .

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth\\_PreferedAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## DroppedFrames

### TVideoGrabber.DroppedFrames

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Current number of dropped frames.

#### Declaration

**property** DroppedFrames: LongInt **read** GetDroppedFrameCount;

\_\_property **int** DroppedFrames=read=GetDroppedFrameCount, **nodefault**

Property DroppedFrames As Long

#### Description

Used to retrieve the current number of dropped frames since [preview](#) or [recording](#) started.

## See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

## DroppedFramesPollingInterval

### TVideoGrabber.DroppedFramesPollingInterval

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the polling interval of dropped frames

#### Declaration

**property** DroppedFramesPollingInterval: LongInt **read** GetDroppedFramesPollingInterval **write** SetDroppedFramesPollingInterval **default** DEF\_DroppedFramesPollingInterval;

\_\_property **int** DroppedFramesPollingInterval=read=GetDroppedFramesPollingInterval, write=SetDroppedFramesPollingInterval, **default**=10

Property DVReduceFrameRate As Long

#### Description

Used to specify the polling interval of the dropped frames count, expressed in seconds.

Remarks:

**DroppedFramesPollingInterval = -1** reads the dropped frames count each time a new frame is received (default value)

**DroppedFramesPollingInterval = 0** disables the polling of dropped frames

E.g. DroppedFramesPollingInterval = 30 polls the dropped frames count every 30 seconds.

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## DualDisplay\_Active

### TVideoGrabber.DualDisplay\_Active

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_Active](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_Active: Boolean **index** vwActive **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_Active;

\_\_property **bool** DualDisplay\_Active=read=GetDisplay2BoolProperties, write=SetDisplay2BoolProperties, index=0, **default**=0

Property DualDisplay\_Active As Boolean

#### Description

Same as [Display\\_Active](#) , but concerns the 2nd video window.

## See Also

[Dual display TVideoRenderer](#)
[AdjustPixelAspectRatio](#)
[Display](#)
[Active](#)
[Display](#)
[AlphaBlendEnabled](#)
[Display](#)
[AlphaBlendValue](#)
[Display](#)
[AutoSize](#)
[Display](#)
[Embedded](#)
[Display](#)
[FullScreen](#)
[Display](#)
[Height](#)
[Display](#)
[Left](#)
[Display](#)
[Monitor](#)
[Display](#)
[MouseMovesWindow](#)
[Display](#)
[PanScanRatio](#)
[Display](#)
[SetLocation](#)
[Display](#)
[StayOnTop](#)
[Display](#)
[Top](#)
[Display](#)
[TransparentColorEnabled](#)
[Display](#)
[TransparentColorValue](#)
[Display](#)
[VideoHeight](#)
[Display](#)
[VideoPortEnabled](#)
[Display](#)
[VideoWidth](#)
[Display](#)
[VideoWindowHandle](#)
[Display](#)
[Width](#)
[DualDisplay](#)
[AlphaBlendEnabled](#)
[DualDisplay](#)
[AlphaBlendValue](#)
[DualDisplay](#)
[AutoSize](#)
[DualDisplay](#)
[Embedded](#)
[DualDisplay](#)
[FullScreen](#)
[DualDisplay](#)
[Height](#)
[DualDisplay](#)
[Left](#)
[DualDisplay](#)
[Monitor](#)
[DualDisplay](#)
[MouseMovesWindow](#)
[DualDisplay](#)
[PanScanRatio](#)
[DualDisplay](#)
[SetLocation](#)
[DualDisplay](#)
[StayOnTop](#)
[DualDisplay](#)
[Top](#)
[DualDisplay](#)
[TransparentColorEnabled](#)
[DualDisplay](#)
[TransparentColorValue](#)
[DualDisplay](#)
[VideoHeight](#)
[DualDisplay](#)
[VideoPortEnabled](#)
[DualDisplay](#)
[VideoWidth](#)
[DualDisplay](#)
[VideoWindowHandle](#)
[DualDisplay](#)
[Visible](#)
[DualDisplay](#)
[Width](#)
[IsVideoPortAvailable](#)
[Monitor](#)
[Primary](#)
[Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

## DualDisplay\_AlphaBlendEnabled

### TVideoGrabber.DualDisplay\_AlphaBlendEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Same as [Display\\_AlphaBlendEnabled](#), for the 2nd video window.

#### Declaration

**property** DualDisplay\_AlphaBlendEnabled: Boolean **index** vwAlphaBlendEnabled **read**  
 GetDualDisplayBoolProperties **write** SetDualDisplayBoolProperties **default**  
 DEF\_DualDisplay\_AlphaBlendEnabled;

**\_\_property bool** DualDisplay\_AlphaBlendEnabled=read=GetDualDisplayBoolProperties,  
 write=SetDualDisplayBoolProperties, index=0, **default**=1

Property DualDisplay\_AlphaBlendEnabled As Boolean

#### Description

Same as [Display\\_AlphaBlendEnabled](#), for the 2nd video window.

#### See Also

[Dual display TVideoRenderer](#)
[AdjustPixelAspectRatio](#)
[Display](#)
[Active](#)
[Display](#)
[AlphaBlendEnabled](#)
[Display](#)
[AlphaBlendValue](#)
[Display](#)
[AutoSize](#)
[Display](#)
[Embedded](#)
[Display](#)
[FullScreen](#)
[Display](#)
[Height](#)
[Display](#)
[Left](#)
[Display](#)
[Monitor](#)
[Display](#)
[MouseMovesWindow](#)
[Display](#)
[PanScanRatio](#)
[Display](#)
[SetLocation](#)
[Display](#)
[StayOnTop](#)
[Display](#)
[Top](#)
[Display](#)
[TransparentColorEnabled](#)
[Display](#)
[TransparentColorValue](#)
[Display](#)
[VideoHeight](#)
[Display](#)
[VideoPortEnabled](#)
[Display](#)
[VideoWidth](#)
[Display](#)
[VideoWindowHandle](#)
[Display](#)
[Width](#)
[DualDisplay](#)
[Active](#)
[DualDisplay](#)
[AlphaBlendValue](#)
[DualDisplay](#)
[AutoSize](#)
[DualDisplay](#)
[Embedded](#)
[DualDisplay](#)
[FullScreen](#)
[DualDisplay](#)
[Height](#)
[DualDisplay](#)
[Left](#)
[DualDisplay](#)
[Monitor](#)
[DualDisplay](#)
[MouseMovesWindow](#)
[DualDisplay](#)
[PanScanRatio](#)
[DualDisplay](#)
[SetLocation](#)
[DualDisplay](#)
[StayOnTop](#)
[DualDisplay](#)
[Top](#)
[DualDisplay](#)
[TransparentColorEnabled](#)
[DualDisplay](#)
[TransparentColorValue](#)
[DualDisplay](#)
[VideoHeight](#)
[DualDisplay](#)
[VideoPortEnabled](#)
[DualDisplay](#)
[VideoWidth](#)
[DualDisplay](#)
[VideoWindowHandle](#)
[DualDisplay](#)
[Visible](#)
[DualDisplay](#)
[Width](#)
[IsVideoPortAvailable](#)
[Monitor](#)
[Primary](#)
[Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free EBook and documentation generator](#)

## DualDisplay\_AlphaBlendValue

### TVideoGrabber.DualDisplay\_AlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_AlphaBlendValue](#), for the 2nd video window.

#### Declaration

**property** DualDisplay\_AlphaBlendValue: LongInt **index** vwAlphaBlendValue **read** GetDualDisplayLongIntProperties **write** SetDualDisplayLongIntProperties **default** DEF\_DualDisplay\_AlphaBlendValue;

\_\_**property int** DualDisplay\_AlphaBlendValue=read=GetDualDisplayLongIntProperties, write=SetDualDisplayLongIntProperties, index=12, **default**=240

Property DualDisplay\_AlphaBlendValue As Long

#### Description

Same as [Display\\_AlphaBlendValue](#), for the 2nd video window.

#### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## DualDisplay\_AspectRatio

### TVideoGrabber.DualDisplay\_AspectRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_AspectRatio](#) , for the 2nd video window.

#### Declaration

**property** DualDisplay\_AspectRatio: TAspectRatio **index** 1 **read** GetDisplayAspectRatio **write** SetDisplayAspectRatio **default** ar\_Box;

\_\_**property TAspectRatio** DualDisplay\_AspectRatio=read=GetDisplayAspectRatio, write=SetDisplayAspectRatio, index=1, **default**=0

Property DualDisplay\_AspectRatio As TAspectRatio

#### Description

Same as [Display\\_AspectRatio](#) , for the the 2nd video window.

**See Also**

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [SetLocation](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Visible](#) [DualDisplay](#) [Width](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Edit and Export Markdown Documents

**DualDisplay\_AutoSize****TVideoGrabber.DualDisplay\_AutoSize**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_AutoSize](#) , but concerns the 2nd video window.

**Declaration**

**property** DualDisplay\_AutoSize: Boolean **index** vwAutosize **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_AutoSize;

\_\_property **bool** DualDisplay\_AutoSize=read=GetDisplay2BoolProperties, write=SetDisplay2BoolProperties, index=1, **default**=1

Property DualDisplay\_AutoSize As Boolean

**Description**

Same as [Display\\_AutoSize](#) , but concerns the 2nd video window.

**See Also**

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [SetLocation](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Visible](#) [DualDisplay](#) [Width](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Free EPub and documentation generator

**DualDisplay\_Embedded****TVideoGrabber.DualDisplay\_Embedded**[Prev](#)[Next](#)



## TVideoGrabber Properties

Same as [Display\\_Embedded](#) , but concerns the 2nd video window.

### Declaration

**property** DualDisplay\_Embedded: Boolean **index** vwEmbedded **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_Embedded;

\_\_property **bool** DualDisplay\_Embedded=read=GetDisplay2BoolProperties,  
write=SetDisplay2BoolProperties, index=2, **default**=0

Property DualDisplay\_Embedded As Boolean

### Description

Same as [Display\\_Embedded](#) , but concerns the 2nd video window.

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## DualDisplay\_FullScreen

### TVideoGrabber.DualDisplay\_FullScreen

[Prev](#)
[Next](#)

#### TVideoGrabber Properties

Same as [Display\\_FullScreen](#) , but concerns the 2nd video window.

### Declaration

**property** DualDisplay\_FullScreen: Boolean **index** vwFullScreen **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_FullScreen;

\_\_property **bool** DualDisplay\_FullScreen=read=GetDisplay2BoolProperties,  
write=SetDisplay2BoolProperties, index=3, **default**=0

Property DualDisplay\_FullScreen As Boolean

### Description

Same as [Display\\_FullScreen](#) , but concerns the 2nd video window.

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation](#)

[Display\\_StayOnTop](#)
[Display\\_Top](#)
[Display\\_TransparentColorEnabled](#)
[Display\\_TransparentColorValue](#)
[Display\\_VideoHeight](#)
[Display\\_VideoPortEnabled](#)
[Display\\_VideoWidth](#)
[Display\\_VideoWindowHandle](#)
[Display\\_Width](#)
[DualDisplay\\_Active](#)
[DualDisplay\\_AlphaBlendEnabled](#)
[DualDisplay\\_AlphaBlendValue](#)
[DualDisplay\\_AutoSize](#)
[DualDisplay\\_Embedded](#)
[DualDisplay\\_Height](#)
[DualDisplay\\_Left](#)
[DualDisplay\\_Monitor](#)
[DualDisplay\\_MouseMovesWindow](#)
[DualDisplay\\_PanScanRatio](#)
[DualDisplay\\_SetLocation](#)
[DualDisplay\\_StayOnTop](#)
[DualDisplay\\_Top](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_TransparentColorValue](#)
[DualDisplay\\_VideoHeight](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[IsVideoPortAvailable](#)
[Monitor\\_Primary](#)
[Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Documentation Capabilities with HelpNDoc's User-Friendly UI

## DualDisplay\_Height

### TVideoGrabber.DualDisplay\_Height

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_Height](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_Height: LongInt **index** vwHeight **read** GetDisplay2LongIntProperties **write** SetDisplay2LongIntProperties **default** DEF\_DualDisplay\_Height;

**\_property** **int** DualDisplay\_Height=read=GetDisplay2LongIntProperties, write=SetDisplay2LongIntProperties, index=12, **default**=240

Property DualDisplay\_Height As Long

#### Description

Same as [Display\\_Height](#) , but concerns the 2nd video window.

#### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio](#)
[Display\\_Active](#)
[Display\\_AlphaBlendEnabled](#)
[Display\\_AlphaBlendValue](#)
[Display\\_AutoSize](#)
[Display\\_Embedded](#)
[Display\\_FullScreen](#)
[Display\\_Height](#)
[Display\\_Left](#)
[Display\\_Monitor](#)
[Display\\_MouseMovesWindow](#)
[Display\\_PanScanRatio](#)
[Display\\_SetLocation](#)
[Display\\_StayOnTop](#)
[Display\\_Top](#)
[Display\\_TransparentColorEnabled](#)
[Display\\_TransparentColorValue](#)
[Display\\_VideoHeight](#)
[Display\\_VideoPortEnabled](#)
[Display\\_VideoWidth](#)
[Display\\_VideoWindowHandle](#)
[Display\\_Width](#)
[DualDisplay\\_Active](#)
[DualDisplay\\_AlphaBlendEnabled](#)
[DualDisplay\\_AlphaBlendValue](#)
[DualDisplay\\_AutoSize](#)
[DualDisplay\\_Embedded](#)
[DualDisplay\\_FullScreen](#)
[DualDisplay\\_Left](#)
[DualDisplay\\_Monitor](#)
[DualDisplay\\_MouseMovesWindow](#)
[DualDisplay\\_PanScanRatio](#)
[DualDisplay\\_SetLocation](#)
[DualDisplay\\_StayOnTop](#)
[DualDisplay\\_Top](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_TransparentColorValue](#)
[DualDisplay\\_VideoHeight](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[IsVideoPortAvailable](#)
[Monitor\\_Primary](#)
[Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Create High-Quality Documentation with a Help Authoring Tool

## DualDisplay\_Left

### TVideoGrabber.DualDisplay\_Left

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_Left](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_Left: LongInt **index** vwLeft **read** GetDisplay2LongIntProperties **write** SetDisplay2LongIntProperties **default** DEF\_DualDisplay\_Left;

\_\_property **int** DualDisplay\_Left=read=GetDisplay2LongIntProperties, write=SetDisplay2LongIntProperties, index=13, **default**=0

Property DualDisplay\_Left As Long

#### Description

Same as [Display\\_Left](#) , but concerns the 2nd video window.

#### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display\\_Left Display\\_Monitor Display\\_MouseMovesWindow Display\\_PanScanRatio Display\\_SetLocation Display\\_StayOnTop Display\\_Top Display\\_TransparentColorEnabled Display\\_TransparentColorValue Display\\_VideoHeight Display\\_VideoPortEnabled Display\\_VideoWidth Display\\_VideoWindowHandle Display\\_Width DualDisplay\\_Active DualDisplay\\_AlphaBlendEnabled DualDisplay\\_AlphaBlendValue DualDisplay\\_AutoSize DualDisplay\\_Embedded DualDisplay\\_FullScreen DualDisplay\\_Height DualDisplay\\_Monitor DualDisplay\\_MouseMovesWindow DualDisplay\\_PanScanRatio DualDisplay\\_SetLocation DualDisplay\\_StayOnTop DualDisplay\\_Top DualDisplay\\_TransparentColorEnabled DualDisplay\\_TransparentColorValue DualDisplay\\_VideoHeight DualDisplay\\_VideoPortEnabled DualDisplay\\_VideoWidth DualDisplay\\_VideoWindowHandle DualDisplay\\_Visible DualDisplay\\_Width IsVideoPortAvailable Monitor\\_Primary\\_Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Effortlessly create a professional-quality documentation website with HelpNDoc

## DualDisplay\_Monitor

### TVideoGrabber.DualDisplay\_Monitor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_Monitor](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_Monitor: LongInt **index** vwMonitor **read** GetDisplay2LongIntProperties **write** SetDisplay2LongIntProperties **default** DEF\_DualDisplay\_Monitor;

\_\_property **int** DualDisplay\_Monitor=read=GetDisplay2LongIntProperties, write=SetDisplay2LongIntProperties, index=8, **default**=0

Property DualDisplay\_Monitor As Long

#### Description

Same as [Display\\_Monitor](#) , but concerns the 2nd video window.

#### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display\\_Left Display\\_Monitor Display\\_MouseMovesWindow Display\\_PanScanRatio Display\\_SetLocation Display\\_StayOnTop Display\\_Top Display\\_TransparentColorEnabled Display\\_TransparentColorValue Display\\_VideoHeight Display\\_VideoPortEnabled Display\\_VideoWidth Display\\_VideoWindowHandle](#)



[Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_SetLocation](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

## DualDisplay\_MouseMovesWindow

### TVideoGrabber.DualDisplay\_MouseMovesWindow

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_MouseMovesWindow](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_MouseMovesWindow: Boolean **index** vwMouseMovesWindow **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_MouseMovesWindow;

\_\_property **bool** DualDisplay\_MouseMovesWindow=read=GetDisplay2BoolProperties, write=SetDisplay2BoolProperties, index=6, **default**=1

Property DualDisplay\_MouseMovesWindow As Boolean

#### Description

Same as [Display\\_MouseMovesWindow](#) , but concerns the 2nd video window.

#### See Also

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_SetLocation](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## DualDisplay\_PanScanRatio

### TVideoGrabber.DualDisplay\_PanScanRatio

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_AspectRatio](#), but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_PanScanRatio: LongInt **index** vwPanScanRatio **read** GetDisplay2LongIntProperties **write** SetDisplay2LongIntProperties **default** DEF\_Display\_PanScanRatio;

\_\_**property int** DualDisplay\_PanScanRatio==GetDisplay2LongIntProperties,  
write=SetDisplay2LongIntProperties, index=11, **default**=50;

Property DualDisplay\_PanScanRatio as Long

#### Description

Same as [Display\\_AspectRatio](#), but concerns the 2nd video window.

#### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## DualDisplay\_StayOnTop

### TVideoGrabber.DualDisplay\_StayOnTop

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_StayOnTop](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_StayOnTop: Boolean **index** vwStayOnTop **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_StayOnTop;

\_\_**property bool** DualDisplay\_StayOnTop=read=GetDisplay2BoolProperties,  
write=SetDisplay2BoolProperties, index=5, **default**=0

Property DualDisplay\_StayOnTop As Boolean

#### Description

Same as [Display\\_StayOnTop](#) , but concerns the 2nd video window.

#### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue](#)

[DualDisplay\\_AutoSize](#)
[DualDisplay\\_Embedded](#)
[DualDisplay\\_FullScreen](#)
[DualDisplay\\_Height](#)
[DualDisplay\\_Left](#)
[DualDisplay\\_Monitor](#)
[DualDisplay\\_MouseMovesWindow](#)
[DualDisplay\\_PanScanRatio](#)
[DualDisplay\\_SetLocation](#)
[DualDisplay\\_Top](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_TransparentColorValue](#)
[DualDisplay\\_VideoHeight](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[IsVideoPortAvailable](#)
[Monitor\\_Primary\\_Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth\\_PreferedAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## DualDisplay\_Top

### TVideoGrabber.DualDisplay\_Top

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Same as [Display\\_Top](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_Top: LongInt **index** vwTop **read** GetDisplay2LongIntProperties **write** SetDisplay2LongIntProperties **default** DEF\_DualDisplay\_Top;

\_\_property **int** DualDisplay\_Top=read=GetDisplay2LongIntProperties, write=SetDisplay2LongIntProperties, index=14, **default**=0

Property DualDisplay\_Top As Long

#### Description

Same as [Display\\_Top](#) , but concerns the 2nd video window.

#### See Also

[Dual display](#)
[TVideoRenderer\\_AdjustPixelAspectRatio](#)
[Display\\_Active](#)
[Display\\_AlphaBlendEnabled](#)
[Display\\_AlphaBlendValue](#)
[Display\\_AutoSize](#)
[Display\\_Embedded](#)
[Display\\_FullScreen](#)
[Display\\_Height](#)
[Display\\_Left](#)
[Display\\_Monitor](#)
[Display\\_MouseMovesWindow](#)
[Display\\_PanScanRatio](#)
[Display\\_SetLocation](#)
[Display\\_StayOnTop](#)
[Display\\_Top](#)
[Display\\_TransparentColorEnabled](#)
[Display\\_TransparentColorValue](#)
[Display\\_VideoHeight](#)
[Display\\_VideoPortEnabled](#)
[Display\\_VideoWidth](#)
[Display\\_VideoWindowHandle](#)
[Display\\_Width](#)
[DualDisplay\\_Active](#)
[DualDisplay\\_AlphaBlendEnabled](#)
[DualDisplay\\_AlphaBlendValue](#)
[DualDisplay\\_AutoSize](#)
[DualDisplay\\_Embedded](#)
[DualDisplay\\_FullScreen](#)
[DualDisplay\\_Height](#)
[DualDisplay\\_Left](#)
[DualDisplay\\_Monitor](#)
[DualDisplay\\_MouseMovesWindow](#)
[DualDisplay\\_PanScanRatio](#)
[DualDisplay\\_SetLocation](#)
[DualDisplay\\_StayOnTop](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_TransparentColorValue](#)
[DualDisplay\\_VideoHeight](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[IsVideoPortAvailable](#)
[Monitor\\_Primary\\_Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth\\_PreferedAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

## DualDisplay\_TransparentColorEnabled

### TVideoGrabber.DualDisplay\_TransparentColorEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Same as [Display\\_TransparentColorEnabled](#), but for the 2nd video window.

#### Declaration

**property** DualDisplay\_TransparentColorEnabled: Boolean **index** vwTransparentColorEnabled **read** GetDualDisplayBoolProperties **write** SetDualDisplayBoolProperties **default** DEF\_DualDisplay\_TransparentColorEnabled;

**\_\_property bool** DualDisplay\_TransparentColorEnabled=read=GetDualDisplayBoolProperties, write=SetDualDisplayBoolProperties, index=0, **default**=1

Property DualDisplay\_TransparentColorEnabled As Boolean

### Description

Same as [Display\\_TransparentColorEnabled](#), but for the 2nd video window.

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## DualDisplay\_TransparentColorValue

### TVideoGrabber.DualDisplay\_TransparentColorValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Same as [Display\\_TransparentColorValue](#), but for the 2nd video window.

### Declaration

**property** DualDisplay\_TransparentColorValue: LongInt **index** vwTransparentColorValue **read** GetDualDisplayLongIntProperties **write** SetDualDisplayLongIntProperties **default** DEF\_DualDisplay\_TransparentColorValue;

**\_\_property int** DualDisplay\_TransparentColorValue=read=GetDualDisplayLongIntProperties, write=SetDualDisplayLongIntProperties, index=12, **default**=240

Property DualDisplay\_TransparentColorValue As Long

### Description

Same as [Display\\_TransparentColorValue](#), but for the 2nd video window.

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio](#)

[DualDisplay\\_SetLocation](#)
[DualDisplay\\_StayOnTop](#)
[DualDisplay\\_Top](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_VideoHeight](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[IsVideoPortAvailable](#)
[Monitor\\_Primary\\_Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## DualDisplay\_VideoHeight

### TVideoGrabber.DualDisplay\_VideoHeight

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Returns the height of the 2nd video window.

#### Declaration

**property** DualDisplay\_VideoHeight: LongInt **read** GetDualDisplay\_VideoHeight;

\_\_property **int** DualDisplay\_VideoHeight=read=GetDualDisplay\_VideoHeight, **nodefault**

Property DualDisplay\_VideoHeight As Long

#### Description

Used to retrieve the height of the 2nd video window when [dual display](#) is used.

#### See Also

[Dual display](#)
[TVideoRenderer](#)
[AdjustPixelAspectRatio](#)
[Display\\_Active](#)
[Display\\_AlphaBlendEnabled](#)
[Display\\_AlphaBlendValue](#)
[Display\\_AutoSize](#)
[Display\\_Embedded](#)
[Display\\_FullScreen](#)
[Display\\_Height](#)
[Display\\_Left](#)
[Display\\_Monitor](#)
[Display\\_MouseMovesWindow](#)
[Display\\_PanScanRatio](#)
[Display\\_SetLocation](#)
[Display\\_StayOnTop](#)
[Display\\_Top](#)
[Display\\_TransparentColorEnabled](#)
[Display\\_TransparentColorValue](#)
[Display\\_VideoHeight](#)
[Display\\_VideoPortEnabled](#)
[Display\\_VideoWidth](#)
[Display\\_VideoWindowHandle](#)
[Display\\_Width](#)
[DualDisplay\\_Active](#)
[DualDisplay\\_AlphaBlendEnabled](#)
[DualDisplay\\_AlphaBlendValue](#)
[DualDisplay\\_AutoSize](#)
[DualDisplay\\_Embedded](#)
[DualDisplay\\_FullScreen](#)
[DualDisplay\\_Height](#)
[DualDisplay\\_Left](#)
[DualDisplay\\_Monitor](#)
[DualDisplay\\_MouseMovesWindow](#)
[DualDisplay\\_PanScanRatio](#)
[DualDisplay\\_SetLocation](#)
[DualDisplay\\_StayOnTop](#)
[DualDisplay\\_Top](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_TransparentColorValue](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[IsVideoPortAvailable](#)
[Monitor\\_Primary\\_Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[OnLeavingFullScreen](#)
[SetParentWindow](#)
[SetWindowTransparency](#)
[VideoDoubleBuffered](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)
[VideoRendererPriority](#)
[VideoVisibleWhenStopped](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## DualDisplay\_VideoPortEnabled

### TVideoGrabber.DualDisplay\_VideoPortEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Same as [Display\\_VideoPortEnabled](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_VideoPortEnabled: Boolean **index** vwVideoPortEnabled **read**

GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_VideoPortEnabled;

\_\_property **bool** DualDisplay\_VideoPortEnabled=read=GetDisplay2BoolProperties,  
write=SetDisplay2BoolProperties, index=7, **default**=1



Property DualDisplay\_VideoPortEnabled As Boolean

### Description

Same as [Display\\_VideoPortEnabled](#) , but concerns the 2nd video window.

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Streamline Your Documentation Process with HelpNDoc's Project Analyzer

## DualDisplay\_VideoWidth

### TVideoGrabber.DualDisplay\_VideoWidth

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the width of the 2nd video window.

### Declaration

**property** DualDisplay\_VideoWidth: LongInt **read** GetDualDisplay\_VideoWidth;

**\_\_property** **int** DualDisplay\_VideoWidth=read=GetDualDisplay\_VideoWidth, **nodefault**

Property Display\_VideoWidth As Long

### Description

Used to retrieve the width of the 2nd video window when [dual display](#) is used.

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

## DualDisplay\_VideoWindowHandle

### TVideoGrabber.DualDisplay\_VideoWindowHandle

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Returns the window handle of the 2nd video window.

#### Declaration

**property** DualDisplay\_VideoWindowHandle: Hwnd **read** GetDualDisplay\_VideoWindowHandle;

\_\_property Hwnd DualDisplay\_VideoWindowHandle=read=GetDualDisplay\_VideoWindowHandle, **nodefault**

Property Display\_VideoWindowHandle As Long

#### Description

Used to retrieve the window handle of the 2nd video window.  
Returns 0 if the 2nd video window does not exist.

#### See Also

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [SetLocation](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [Visible](#) [DualDisplay](#) [Width](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer](#)

## DualDisplay\_Visible

### TVideoGrabber.DualDisplay\_Visible

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Shows / hide the 2nd video window

#### Declaration

**property** DualDisplay\_Visible: Boolean **index** vwVisible **read** GetDisplay2BoolProperties **write** SetDisplay2BoolProperties **default** DEF\_DualDisplay\_Visible;

\_\_property bool DualDisplay\_Visible=read=GetDisplay2BoolProperties, write=SetDisplay2BoolProperties, index=1, **default**=1

Property DualDisplay\_Visible As Boolean

#### Description



Used to show / hide the 2nd video window when the [dual display](#) feature is used.  
Enabled by default.

#### See Also

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [SetLocation](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Width](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Elevate Your CHM Help Files with HelpNDoc's Advanced Customization Options

## DualDisplay\_Width

### TVideoGrabber.DualDisplay\_Width

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Same as [Display\\_Width](#) , but concerns the 2nd video window.

#### Declaration

**property** DualDisplay\_Width: LongInt **index** vwWidth **read** GetDisplay2LongIntProperties **write** SetDisplay2LongIntProperties **default** DEF\_DualDisplay\_Width;

\_\_property **int** DualDisplay\_Width=read=GetDisplay2LongIntProperties, write=SetDisplay2LongIntProperties, index=11, **default**=320

Property DualDisplay\_Width As Long

#### Description

Same as [Display\\_Width](#) , but concerns the 2nd video window.

#### See Also

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [SetLocation](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Visible](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Free EPub producer

## DVDateTimeEnabled

### TVideoGrabber.DVDateTimeEnabled

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Enables the reporting of the DV date/time

#### Declaration

**property** DVDateTimeEnabled: Boolean **read** GetDVDateTimeEnabled **write** SetDVDateTimeEnabled  
**default** DEF\_DVDateTimeEnabled;

\_\_**property bool** DVDateTimeEnabled=read=GetDVDateTimeEnabled, write=SetDVDateTimeEnabled,  
**default=1**

Property DVDateTimeEnabled as Boolean

#### Description

Used to enable/disable the reporting of the date/time for DV sources (e.g. like a camcorder)

This property is enabled by default. Disabling it may save a few CPU load (non-DV sources are not concerned by this setting)

#### See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#)  
[DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#)  
[DVrgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#)  
[OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Keep Your PDFs Safe from Unauthorized Access with These Security Measures](#)

## DVDDiscontinuityMinimumInterval

### TVideoGrabber.DVDDiscontinuityMinimumInterval

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Specifies a minimum interval of time between DV discontinuity notifications.

#### Declaration

**property** DVDDiscontinuityMinimumInterval: LongInt **read** GetDVDDiscontinuityMinimumInterval **write** SetDVDDiscontinuityMinimumInterval **default** DEF\_DVDDiscontinuityMinimumInterval;

\_\_**property int** DVDDiscontinuityMinimumInterval=read=GetDVDDiscontinuityMinimumInterval,  
write=SetDVDDiscontinuityMinimumInterval, **default=3**

Property DVDDiscontinuityMinimumInterval As Long

#### Description

This property specifies a minimum interval of time between the notifications of DV discontinuities detected in the date/time and reported by the [OnDVDDiscontinuity](#) event.

The end of a recording sequence having a duration shorter than this interval will not be notified by the [OnDVDDiscontinuity](#) event.

The value is expressed in seconds.

**Default value:** 3 seconds

**Minimum value:** 1 second

#### See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDDateTimeEnabled](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## DVDTitle

### TVideoGrabber.DVDTitle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the DVD title to play

#### Declaration

**property** DVDTitle: LongInt **read** GetDVDTitle **write** SetDVDTitle **default** DEF\_DVDTitle;

\_\_**property** int DVDTitle==GetDVDTitle, write=SetDVDTitle, **default**=0;

Property DVDTitle as Long

#### Description

Used to specify a DVD title to play, starting from 1.

If DVDTitle = 0 (default) the DVD menu is shown when invoking [OpenPlayer](#).

E.g.:

```
VideoGrabber.PlayerFileName = "E:
VideoGrabber.DVDTitle = 0
VideoGrabber.OpenPlayer()
```

If DVDTitle > 0 the corresponding DVD title is played immediately.

E.g.:

```
VideoGrabber.PlayerFileName = "E:
VideoGrabber.DVDTitle = 1
VideoGrabber.OpenPlayer()
```

#### See Also

[DVDInfo](#) [OpenDVD](#) [PlayerFileName](#)

Created with the Standard Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's HTML5 template](#)

## DVEncoder\_VideoFormat

### TVideoGrabber.DVEncoder\_VideoFormat

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the DV encoding format.

#### Declaration

**property** DVEncoder\_VideoFormat: TDVVideoFormat **read** GetDVEncoder\_VideoFormat **write**

SetDVEncoder\_VideoFormat;

\_\_property TDVVideoFormat DVEncoder\_VideoFormat=read=GetDVEncoder\_VideoFormat,  
write=SetDVEncoder\_VideoFormat, **nodefault**

Property DVEncoder\_VideoFormat As TxDVVideoFormat

### Description

Used to specify the DV encoding format when sending the audio/video streams to a DV device (when [RecordingMethod](#) is rm\_SendToDV).  
The value is a [TDVVideoFormat](#) type.

### See Also

[Send to DV](#) [DVEncoder\\_VideoResolution](#) [DVEncoder\\_VideoStandard](#) [SendToDV\\_DeviceIndex](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## DVEncoder\_VideoResolution

### TVideoGrabber.DVEncoder\_VideoResolution

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the DV encoding resolution.

### Declaration

**property** DVEncoder\_VideoResolution: TDVSize **read** GetDVEncoder\_VideoResolution **write** SetDVEncoder\_VideoResolution;

\_\_property TDVSize DVEncoder\_VideoResolution=read=GetDVEncoder\_VideoResolution,  
write=SetDVEncoder\_VideoResolution, **nodefault**

Property DVEncoder\_VideoResolution As TxDVSize

### Description

Used to specify the DV encoding resolution when sending the audio/video streams to a DV device (when [RecordingMethod](#) is rm\_SendToDV).  
The value is [TDVSize](#) a type.

### See Also

[Send to DV](#) [DVEncoder\\_VideoFormat](#) [DVEncoder\\_VideoStandard](#) [SendToDV\\_DeviceIndex](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

## DVEncoder\_VideoStandard

### TVideoGrabber.DVEncoder\_VideoStandard

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the DV encoding standard.

### Declaration

**property** DVEncoder\_VideoStandard: TDVVideoStandard **read** GetDVEncoder\_VideoStandard **write** SetDVEncoder\_VideoStandard;

\_\_property TDVVideoStandard DVEncoder\_VideoStandard=read=GetDVEncoder\_VideoStandard,  
write=SetDVEncoder\_VideoStandard, **nodefault**

Property DVEncoder\_VideoStandard As TxDVVideoStandard

### Description

Used to specify the DV encoding standard (PAL or NTSC) when sending the audio/video streams to a DV device (when [RecordingMethod](#) is rm\_SendToDV).  
The value is a [TDVVideoStandard](#) type.

### See Also

[Send to DV](#) [DVEncoder\\_VideoFormat](#) [DVEncoder\\_VideoResolution](#) [SendToDV\\_DeviceIndex](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## DVRecordingInNativeFormatSeparatesStreams

### TVideoGrabber.DVRecordingInNativeFormatSeparatesStreams

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specify if the DV audio and video streams must be stored as separated streams.

### Declaration

**property** DVRecordingInNativeFormatSeparatesStreams: Boolean **read**

GetDVRecordingInNativeFormatSeparatesStreams **write** SetDVRecordingInNativeFormatSeparatesStreams  
**default** true;

\_\_property bool

DVRecordingInNativeFormatSeparatesStreams=read=GetDVRecordingInNativeFormatSeparatesStreams,  
write=SetDVRecordingInNativeFormatSeparatesStreams, **default**=1

Property DVRecordingInNativeFormatSeparatesStreams As Boolean

### Description

When recording DV audio+video in native format in an AVI file ([RecordingInNativeFormat](#) enabled and [RecordingMethod](#) = rm\_AVI), this property specifies if the DV audio and video streams must be stored as 2 separated audio and video streams, or as a single A/V interleaved stream:

- if enabled, the DV audio and video are stored within the AVI as 2 separated streams: a video stream of type "dvsd", and an audio stream of type "auds".

- if disabled the DV audio and video are stored as a single interleaved audio and video stream of type "iavs".

### See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#)  
[DVDDiscontinuityMinimumInterval](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#)  
[IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

## DVReduceFrameRate

### TVideoGrabber.DVReduceFrameRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Reduces the frame rate of the DV video stream.

### Declaration

**property** DVReduceFrameRate: Boolean **read** GetDVReduceFrameRate **write** SetDVReduceFrameRate  
**default** DEF\_DVReduceFrameRate;

\_\_property **bool** DVReduceFrameRate=read=GetDVReduceFrameRate, write=SetDVReduceFrameRate,  
**default=0**

Property DVReduceFrameRate As Boolean

### Description

Used to reduce the frame rate of the DV video stream.

If enabled, half of the frames are discarded in the video stream:

- for NTSC, the frame rate is reduced from 30 frames per second (fps) to 15 fps,
- for PAL, the frame rate is reduced from 25 fps to 12.5 fps.

This property can be used only if the current video capture device source is DV ([IsDigitalVideoIn](#) returns true).

### See Also

[Frame rate](#) [TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#)  
[DVDateTimeEnabled](#) [DVD discontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#)  
[DVRgb219](#) [DVTimeCodeEnabled](#) [FrameRate](#) [FramerateDivider](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#)  
[IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## DVRgb219

### TVideoGrabber.DVRgb219

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Disables the DV color range stretching.

### Declaration

**property** DVRgb219: Boolean **read** GetDVRgb219 **write** SetDVRgb219 **default** DEF\_DVRgb219;

\_\_property **bool** DVRgb219=read=GetDVRgb219, write=SetDVRgb219, **default=0**

Property DVRgb219 As Boolean

### Description

Used to disable the color range stretching of DV video streams going through the DV decoder.

The DV video format has a dynamic range of 16-235. By default, when the DV Video Decoder decodes to 24-bit or 32-bit RGB, by stretching the color range from 0-255.

In RGB-219 mode, the decoder does not stretch the color range.

This property can be used only if the current video capture device source is DV ([IsDigitalVideoIn](#) returns true).

### See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#)  
[DVD discontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#)  
[DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#)  
[OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

## DVTimeCodeEnabled

### TVideoGrabber.DVTimeCodeEnabled

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Enables/disable the reporting of the DV TimeCode

### Declaration

**property** DVTimeCodeEnabled: Boolean **read** GetDVTTimeCodeEnabled **write** SetDVTTimeCodeEnabled  
**default** DEF\_DVTTimeCodeEnabled;

**\_\_property** bool DVTimeCodeEnabled=read=GetDVTTimeCodeEnabled, write=SetDVTTimeCodeEnabled,  
**default**=1

Property DVTimeCodeEnabled as Boolean

### Description

Used to enable/disable the reporting of the DV TimeCode for DV sources (e.g. like a camcorder)

This property is enabled by default. Disabling it may save a few CPU load (non-DV sources are not concerned by this setting)

### See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDDateTimeEnabled](#)  
[DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#)  
[DVRgb219](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#)  
[OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## EncryptionMethod

### [TVideoGrabber](#).EncryptionMethod

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Specifies the encryption method

### Declaration

**property** EncryptionMethod: TEncryptionMethod **read** GetEncryptionMethod **write** SetEncryptionMethod  
**default** DEF\_EncryptionMethod;

**\_\_property** TEncryptionMethod EncryptionMethod=read=GetEncryptionMethod,  
write=SetEncryptionMethod, **default**=0

### Description

Specifies the encryption method enabled by [SetEncryptionKey](#)

### See Also

[TEncryptionMethod](#) [Decrypt](#) [File](#) [Encrypt](#) [File](#) [SetDecryptionKey](#) [SetEncryptionKey](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## EventNotificationSynchronone

### [TVideoGrabber](#).EventNotificationSynchronone

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Speficies if the events are notified synchronously.

### Declaration

**property** EventNotificationSynchronone: Boolean **read** GetEventNotificationSynchronone **write**



SetEventNotificationSynchronone **default** DEF\_EventNotificationSynchronone;

\_\_property **bool** EventNotificationSynchronone=read=GetEventNotificationSynchronone,  
write=SetEventNotificationSynchronone, **default**=0

Property EventNotificationSynchronone As Boolean

### Description

Used to specify if the events occur synchronously.

If you invoke TVideoGrabber from a thread you can disable this property.

See also [SyncCommands](#).

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## ExtraDLLPath

### TVideoGrabber.ExtraDLLPath

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Path of the binaries of the external Datastead DirectShow filters

### Declaration

**property** ExtraDLLPath: **string** read GetExtraDLLPath **write** SetExtraDLLPath;

\_\_property wchar\_t \*ExtraDLLPath==GetExtraDLLPath, write=SetExtraDLLPath;

### Description

Specifies the location of the x86/x64 folders of the binaries of the Datastead DirectShow filters. It is possible to invoke the [Datastead RTSP/RTMP/HTTP/ONVIF DirectShow source filter](#) or [Datastead Multipurpose Encoder](#) from the TVideoGrabber SDK just by specifying to TVideoGrabber the location of the binaries, **without having to run the .exe installer, and without having to register the filter(s) with regsvr32.exe**.

Just copy the x86 and x64 folders (containing the binaries of the filter) to a folder of your choice, and set ExtraDLLPath = this folder.

E.g. if the "x86" and "x64" folders have been copied to c:/datasteadfilter, set VideoGrabber.ExtraDLLPath = "c:/datasteadfilter"

Example:

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.IPCameraURL = "rtsp://..."
VideoGrabber.ExtraDLLPath = "c:/datasteadfilter"
VideoGrabber.StartPreview();
```

It is also possible to specify a different DLL path per filter, by adding the **[RTSP]** or **[MPE]** prefix at the beginning, e.g.:

```
VideoGrabber.ExtraDLLPath = "[RTSP]c:/RTSPfolder1
VideoGrabber.ExtraDLLPath = "[MPE]c:/MPEfolder2
```

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#)  
[TONONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#)  
[GetLastErrorMessages](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#)  
[ONONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#)  
[ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras\\_IPRange](#)  
[ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#)  
[ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#)

[ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#)  
[OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Easily Add Encryption and Password Protection to Your PDFs](#)

## FixFlickerOrBlackCapture

### TVideoGrabber.FixFlickerOrBlackCapture

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to fix display problems with some video capture devices.

#### Declaration

**property** FixFlickerOrBlackCapture: Boolean **read** GetFixFlickerOrBlackCapture **write** SetFixFlickerOrBlackCapture **default** DEF\_FixFlickerOrBlackCapture;

\_\_property **bool** FixFlickerOrBlackCapture=read=GetFixFlickerOrBlackCapture, write=SetFixFlickerOrBlackCapture, **default**=0

Property FixFlickerOrBlackCapture As Boolean

#### Description

This property is used to fix the possible following problems with some video capture devices:

- flickering,
- black frame capture when invoking CaptureFrameTo,
- excessive apparent frame rate.

Try to enable this property if you get this kind of problem.

Enabling this property requires a bit more CPU, so it should be kept enabled **only** if it fixes these kind of problems.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

## FrameCaptureHeight

### TVideoGrabber.FrameCaptureHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies a height for the captured frame.

#### Declaration

**property** FrameCaptureHeight: LongInt **read** GetFrameCaptureHeight **write** SetFrameCaptureHeight **default** DEF\_FrameCaptureHeight;

\_\_property **int** FrameCaptureHeight==GetFrameCaptureHeight, write=SetFrameCaptureHeight, **default**=-1;

Property FrameCaptureHeight as Long

## Description

Used to specify a height for the captured frame when capturing frames with [CaptureFrameTo](#) or the [burst mode](#).

Default value = -1 (disabled)

If a value > 0 is specified, the captured frame will be stretched to the specified height, otherwise the height of the video frame will be used.

## See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## FrameCaptureWidth

### TVideoGrabber.FrameCaptureWidth

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies a width for the captured frame.

## Declaration

property FrameCaptureWidth: LongInt read GetFrameCaptureWidth write SetFrameCaptureWidth default DEF\_FrameCaptureWidth;

\_\_property int FrameCaptureWidth==GetFrameCaptureWidth, write=SetFrameCaptureWidth, default=-1;

Property FrameCaptureWidth as Long

## Description

Used to specify a width for the captured frame when capturing frames with [CaptureFrameTo](#) or the [burst mode](#).

Default value = -1 (disabled)

If a value > 0 is specified, the captured frame will be stretched to the specified width, otherwise the width of the video frame will be used.

## See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## FrameCaptureWithoutOverlay

### TVideoGrabber.FrameCaptureWithoutOverlay

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies whether [frame capture](#) functions must capture unmodified frames when drawing over video frames.

#### Declaration

**property** FrameCaptureWithoutOverlay: Boolean **read** GetFrameCaptureWithoutOverlay **write** SetFrameCaptureWithoutOverlay **default** DEF\_FrameCaptureWithoutOverlay;

\_\_property **bool** FrameCaptureWithoutOverlay=read=GetFrameCaptureWithoutOverlay, write=SetFrameCaptureWithoutOverlay, **default**=0

Property FrameCaptureWithoutOverlay As Boolean

#### Description

Used to specify whether [frame capture](#) must be performed before drawing over video frames (unmodified video frames) or after drawing over video frames (modified video frames).

- if FrameCaptureBeforeDrawing is disabled (default), frame overlay affects the captured frames as well as preview and recording files.

-if FrameCaptureBeforeDrawing is enabled, the frames captured by [CaptureFrameTo](#) or [in burst mode](#) concerns unmodified native frames.

In this case text, shapes, lines or bitmap drawn over video frames appear only on the preview window or recording files.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

## FrameCaptureZoomSize

### TVideoGrabber.FrameCaptureZoomSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the size of the captured video frames, in percent of the [video source size](#) .

#### Declaration

**property** FrameCaptureZoomSize: LongInt **read** GetFrameCaptureZoomSize **write** SetFrameCaptureZoomSize **default** DEF\_FrameCaptureZoomSize;

\_\_property **int** FrameCaptureZoomSize=read=GetFrameCaptureZoomSize, write=SetFrameCaptureZoomSize, **default**=100

Property FrameCaptureZoomSize As Long

#### Description

Used modify the size of video frames captured using the [frame capture](#) .

The captured frames will be stretched to the specified percentage of the [video source width](#) and [video source height](#) .

Default value: 100 (percent).

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#)

[BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## FrameGrabber

### TVideoGrabber.FrameGrabber

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables and configure or disables the frame grabber, that allows to capture frames and draw over frames.

#### Declaration

**property** FrameGrabber: TFrameGrabber **read** GetFrameGrabber **write** SetFrameGrabber **default** DEF\_FrameGrabber;

\_\_property TFrameGrabber FrameGrabber=read=GetFrameGrabber, write=SetFrameGrabber, **default**=0

Property FrameGrabber As TFrameGrabber

#### Description

Used to enable or disable the frame grabber.

#### Values:

**fg\_BothStreams**: the frame grabber is applied on both preview stream and capture streams,

**fg\_PreviewStream**: the frame grabber is applied only on the preview stream,

**fg\_CaptureStream**: the frame grabber is applied only on the capture stream,

**fg\_Disabled**: the frame grabber is disabled.

#### Overview

The frame grabber is a RGB filter that lets you capture frames and/or perform graphic and text overlays.

The frame grabber allows to:

- capture video frames: see the [Frame capture](#) ,
- draw (or write text) over video frames: see [Drawing over video frames](#) .

- if **FrameGrabber = fg\_BothStreams**, the overlays are applied to the video window (preview) as well as to the captured frames and the video clip being recorded)

- if **FrameGrabber = fg\_PreviewStream**, the overlays are applied only on the video window (but not on the captured frames or the video clip being recorded)

- if **FrameGrabber = fg\_CaptureStream**, the overlay are applied to the captured frames and the video clip being recorded, but are not visible on the video window.

If you want to capture frames without applying the overlays (but you want the overlays in the preview and/or the recorded video clip), enable the [FrameCaptureWithoutOverlay](#) property.

#### Recording

When it is used on a stream (preview, capture or both), this stream is converted into RGB format.

By default the recording is made in the native format of the video source when [RecordingInNativeFormat](#) is enabled (default)

If you need to apply the overlays to the recorded clip, disable the [RecordingInNativeFormat](#). Then the recording will performed in uncompressed RGB with the overlays applied.

You can compress the video stream with the overlays applied by selecting an audio or video compressor, according to the [CompressionMode](#) and [CompressionType](#) settings.

During recording, the frame grabber can be applied on the recording stream, the preview stream, or both. If you want to preserve the recording format and need however to capture frames, you can either:

- apply the frame grabber only on the preview stream (FrameGrabber = fg\_PreviewStream).
- enable the [RecordingInNativeFormat](#) that will save the clip in the native (unmodified) format coming out of the video source.

Note: using the frame grabber requires more CPU, so it should be disabled if it is not required to capture video frames or draw over them during recording.

#### See Also

[TFrameGrabberRGBFormat](#) [FrameGrabberCurrentRGBFormat](#) [FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#) [GetFrameInfoString](#) [InFrameProgressEvent](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Add an Extra Layer of Security to Your PDFs with Encryption](#)

## FrameGrabberCurrentRGBFormat

### TVideoGrabber.FrameGrabberCurrentRGBFormat

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the current RGB format of the frame grabber.

#### Declaration

**property** FrameGrabberCurrentRGBFormat: TFrameGrabberRGBFormat **read** GetFrameGrabberCurrentRGBFormat;

**\_\_property** TFrameGrabberRGBFormat FrameGrabberCurrentRGBFormat=read=GetFrameGrabberCurrentRGBFormat, **nodefault**

Property FrameGrabberCurrentRGBFormat As TFrameGrabberRGBFormat

#### Description

Used to retrieve the current RGB format of the frame grabber. Useful mainly when the frame grabber has no specified RGB format ([FrameGrabberRGBFormat](#) = fgf\_Default), to retrieve the current RGB format in use after invoking [StartPreview](#), [StartRecording](#) or [OpenPlayer](#).

#### See Also

[TFrameGrabberRGBFormat](#) [FrameGrabber](#) [FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#) [GetFrameInfoString](#) [InFrameProgressEvent](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

## FrameGrabberRGBFormat

### TVideoGrabber.FrameGrabberRGBFormat

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Format of the video frames captured by [CaptureFrameTo](#) or in [burst mode](#) .

#### Declaration

**property** FrameGrabberRGBFormat: TFrameGrabberRGBFormat **read** GetFrameGrabberRGBFormat **write** SetFrameGrabberRGBFormat **default** DEF\_FrameGrabberRGBFormat;

\_\_property TFrameGrabberRGBFormat FrameGrabberRGBFormat=read=GetFrameGrabberRGBFormat, write=SetFrameGrabberRGBFormat, **default**=0

Property FrameGrabberRGBFormat As TFrameGrabberRGBFormat

#### Description

Used to specify the RGB format (see [TFrameGrabberRGBFormat](#)) of the frame grabber. This is the RGB format used for frame capture, and also for text and graphics overlays. the format is fgf\_Default (you let TVideoGrabber choose the best RGB format), the real RGB format in use will be returned by the [FrameGrabberCurrentRGBFormat](#) property.

#### See Also

[TFrameGrabberRGBFormat](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#) [GetFrameInfoString](#) [InFrameProgressEvent](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## FrameNumberStartsFromZero

### TVideoGrabber.FrameNumberStartsFromZero

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies if the frame counting must start from 0

#### Declaration

**property** FrameNumberStartsFromZero: Boolean **read** GetFrameNumberStartsFromZero **write** SetFrameNumberStartsFromZero **default** DEF\_FrameNumberStartsFromZero;

\_\_**property bool** FrameNumberStartsFromZero=read=GetFrameNumberStartsFromZero, write=SetFrameNumberStartsFromZero, **default**=0

property FrameNumberStartsFromZero As Boolean

#### Description

Used to specify if the frame counting must start from 0.

(disabled by default, in this case the 1st frame number is 1)

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## FrameRate

### TVideoGrabber.FrameRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Desired frame rate, expressed in frame per second.

#### Declaration

**property** FrameRate: Double **read** GetFrameRate **write** SetFrameRate;

\_\_**property double** FrameRate=read=GetFrameRate, write=SetFrameRate

Property FrameRate As Double

#### Description

Used to set or retrieve the desired frame rate, expressed in frames per second.



If FrameRate = 0, the default frame rate of the video capture device is used.  
This property does not apply to DV (see the [Frame rate](#) chapter).  
The effective frame rate (for analog as well as DV devices) can be retrieved by using the [CurrentFrameRate](#) property.

#### See Also

[Frame rate](#) [DVReduceFrameRate](#) [FramerateDivider](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Documents to eBooks: A Step-by-Step Guide with HelpNDoc](#)

## FramerateDivider

### TVideoGrabber.FramerateDivider

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Divides the frame rate by the value specified

#### Declaration

**property** FrameRateDivider: LongInt **read** GetFrameRateDivider **write** SetFrameRateDivider **default** DEF\_FrameRateDivider;

**\_\_property** System::LongInt FrameRateDivider==GetFrameRateDivider, write=SetFrameRateDivider, default=0;

#### Description

Divides the frame rate by the value specified.  
This property requires the [frame grabber](#) to be enabled.

Example ("f" = frame delivered):

```
FrameRateDivider = 1: f f f f f f f f f f f f f f f
FrameRateDivider = 2: f . f . f . f . f . f . f . f
FrameRateDivider = 3: f . . f . . f . . f . . f . .
```

#### See Also

[Frame rate](#) [TFrameGrabberRGBFormat](#) [DVReduceFrameRate](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#) [FrameGrabberRGBFormat](#) [FrameRate](#) [GetFrameInfo](#) [GetFrameInfoString](#) [InFrameProgressEvent](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

## GetLastFrameWaitTimeoutMs

### TVideoGrabber.GetLastFrameWaitTimeoutMs

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

When specified the GetLastFrames... functions wait for the next frame

#### Declaration

**property** GetLastFrameWaitTimeoutMs: LongInt **read** GetGetLastFrameWaitTimeoutMs **write** SetGetLastFrameWaitTimeoutMs;

**\_\_property** int ZoomYCenter=read=GetGetLastFrameWaitTimeoutMs, write=SetGetLastFrameWaitTimeoutMs, **nodefault**

Property GetLastFrameWaitTimeoutMs As Long

#### Description

By default the `GetLastFrame...` functions return the current video frame.

When `GetLastFrameWaitTimeoutMs` is set to a duration expressed in milliseconds (e.g. 50), the `GetLastFrame...` functions wait for the next frame, eventually return it, or fail if the timeout specified expires.

- for live sources the maximum duration is not critical, it is possible to set e.g. 1000 milliseconds
- during playback, when paused, we recommend to set a maximum value of 1.5 times the average time between 2 frames, to be sure to catch the next frame but avoid a potential deadlock when the clip is paused and/or when seeking.

E.g. at 30 fps the average time between 2 frames is 33ms, so `GetLastFrameWaitTimeoutMs` = 50ms should work.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## HoldRecording

### TVideoGrabber.HoldRecording

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Retains the beginning of the recording.

#### Declaration

**property** HoldRecording: Boolean **read** GetHoldRecording **write** SetHoldRecording **default** DEF\_HoldRecording;

\_\_property **bool** HoldRecording=read=GetHoldRecording, write=SetHoldRecording, **default**=0

Property HoldRecording As Boolean

#### Description

Used to start the recording in preview mode and retain the beginning of AVI writing.

When **HoldRecording** is set to true before calling [StartRecording](#), the recording graph is build and started in preview mode. Then, as soon as [ResumeRecording](#) is invoked, AVI writing begins.

When calling `StartRecording` directly without using **HoldRecording**, a few seconds are necessary to build the recording graph, so it is not possible to have the AVI file starting at the moment where [StartRecording](#) was called.

By enabling **HoldRecording** before calling [StartRecording](#), the recording is started in preview mode, and the AVI writing begins exactly when [ResumeRecording](#) is invoked.

After invoking `StartRecording`, the [OnRecordingReadyToStart](#) event occurs to let you know that you can invoke [ResumeRecording](#) to start the recording.

From this event the recording can be cancelled by invoking [StopRecording](#).

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#)

[OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Make Help Documentation a Breeze with a Help Authoring Tool](#)

## ImageOverlay\_AlphaBlend

### TVideoGrabber.ImageOverlay\_AlphaBlend

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the alpha blending of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_AlphaBlend: Boolean **read** GetImageOverlay\_AlphaBlend **write** SetImageOverlay\_AlphaBlend **default** DEF\_ImageOverlay\_AlphaBlend;

**\_\_property** bool ImageOverlay\_AlphaBlend==GetImageOverlay\_AlphaBlend, write=SetImageOverlay\_AlphaBlend, **default**=0;

property ImageOverlay\_AlphaBlend as Boolean

#### Description

Enables the alpha blending of an overlaid image.

The alpha blending value must be specified with [ImageOverlay\\_AlphaBlendValue](#)

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Publish Your Word Document as an eBook](#)

## ImageOverlay\_AlphaBlendValue

### TVideoGrabber.ImageOverlay\_AlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the the alpha blending value of the current image overlay selected by [ImageOverlaySelector](#)

**Declaration**

**property** ImageOverlay\_AlphaBlendValue: LongInt **read** GetImageOverlay\_AlphaBlendValue **write** SetImageOverlay\_AlphaBlendValue **default** DEF\_ImageOverlay\_AlphaBlendValue;

**\_\_property int** ImageOverlay\_AlphaBlendValue==GetImageOverlay\_AlphaBlendValue,  
write=SetImageOverlay\_AlphaBlendValue, **default**=180;

property ImageOverlay\_AlphaBlendValue as Long

**Description**

Specifies the alpha blending value of an overlaid image.

The alpha blending must be enabled by [ImageOverlay\\_AlphaBlend](#)

Look at the [Image Overlays](#) chapter for more information.

**See Also**

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Documentation Output with HelpNDoc's  
Advanced Project Analyzer

**ImageOverlay\_ChromaKey****TVideoGrabber.ImageOverlay\_ChromaKey**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the chroma key feature of the current image overlay selected by [ImageOverlaySelector](#)

**Declaration**

**property** ImageOverlay\_ChromaKey: Boolean **read** GetImageOverlay\_ChromaKey **write** SetImageOverlay\_ChromaKey **default** DEF\_ImageOverlay\_ChromaKey;

**\_\_property bool** ImageOverlay\_ChromaKey==GetImageOverlay\_ChromaKey,  
write=SetImageOverlay\_ChromaKey, **default**=0;

property ImageOverlay\_ChromaKey as Boolean

**Description**

Used to enable or disable the chroma key feature of the image overlay.

See the [Chroma Key](#) chapter.

**See Also**

[Chroma key](#) [ImageOverlay\\_ChromaKeyLeewayPercent](#) [ImageOverlay\\_ChromaKeyRGBColor](#)

Created with the Standard Edition of HelpNDoc: Transform Your Word Doc into a Professional-Quality  
eBook with HelpNDoc

## ImageOverlay\_ChromaKeyLeewayPercent

### TVideoGrabber.ImageOverlay\_ChromaKeyLeewayPercent

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the percentage of leeway of the chroma key of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_ChromaKeyLeewayPercent: LongInt **read**  
 GetImageOverlay\_ChromaKeyLeewayPercent **write** SetImageOverlay\_ChromaKeyLeewayPercent **default**  
 DEF\_ImageOverlay\_ChromaKeyLeewayPercent;

#### \_\_property int

ImageOverlay\_ChromaKeyLeewayPercent==GetImageOverlay\_ChromaKeyLeewayPercent,  
 write=SetImageOverlay\_ChromaKeyLeewayPercent, **default**=25;

property ImageOverlay\_ChromaKeyLeewayPercent as Long

#### Description

Used to Specify the percentage of leeway of the chroma key.

See the [Chroma Key](#) chapter.

#### See Also

[Chroma key](#) [ImageOverlay\\_ChromaKey](#) [ImageOverlay\\_ChromaKeyRGBColor](#)

Created with the Standard Edition of HelpNDoc: [Protect Your Confidential PDFs with These Simple Security Measures](#)

## ImageOverlay\_ChromaKeyRGBColor

### TVideoGrabber.ImageOverlay\_ChromaKeyRGBColor

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the RGB color used as chroma key for the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_ChromaKeyRGBColor: LongInt **read** GetImageOverlay\_ChromaKeyRGBColor  
**write** SetImageOverlay\_ChromaKeyRGBColor **default** DEF\_ImageOverlay\_ChromaKeyRGBColor;

**\_\_property int** ImageOverlay\_ChromaKeyRGBColor==GetImageOverlay\_ChromaKeyRGBColor,  
 write=SetImageOverlay\_ChromaKeyRGBColor, **default**=0;

property ImageOverlay\_ChromaKeyRGBColor as Long

#### Description

Specifies the RGB color used as chroma key.

See the [Chroma Key](#) chapter.

#### See Also

[Chroma key](#) [ImageOverlay\\_ChromaKey](#) [ImageOverlay\\_ChromaKeyLeewayPercent](#)

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## ImageOverlay\_Height

### TVideoGrabber.ImageOverlay\_Height

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the height of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_Height: LongInt **read** GetImageOverlay\_Height **write** SetImageOverlay\_Height  
**default** DEF\_ImageOverlay\_Height;

**\_\_property int** ImageOverlay\_Height==GetImageOverlay\_Height, write=SetImageOverlay\_Height,  
**default=-1;**

property ImageOverlay\_Height as Long

#### Description

Specifies the height to which the overlayed image will be stretched.

Set ImageOverlay\_Height = -1 to use the real height of the loaded image.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#)  
[ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_TopLocation](#)  
[ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#)  
[ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Free Qt Help documentation generator](#)

## ImageOverlay\_LeftLocation

### TVideoGrabber.ImageOverlay\_LeftLocation

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the left location of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_LeftLocation: LongInt **read** GetImageOverlay\_LeftLocation **write**  
SetImageOverlay\_LeftLocation **default** DEF\_ImageOverlay\_LeftLocation;

**\_\_property int** ImageOverlay\_LeftLocation==GetImageOverlay\_LeftLocation,  
write=SetImageOverlay\_LeftLocation, **default=10;**

property ImageOverlay\_LeftLocation as Long

#### Description

Specifies the left location of the overlayed image over the video frames.



The value must be between 0 and the current video width.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## ImageOverlay\_RotationAngle

### TVideoGrabber.ImageOverlay\_RotationAngle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies a rotation angle for the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_RotationAngle: Double **read** GetImageOverlay\_RotationAngle **write** SetImageOverlay\_RotationAngle **default** DEF\_ImageOverlay\_RotationAngle

**\_\_property double** ImageOverlay\_RotationAngle==GetImageOverlay\_RotationAngle, write=SetImageOverlay\_RotationAngle, **default**=0;

**property** \_ImageOverlay\_RotationAngle as Double

#### Description

Used to specify a rotation angle for the current image overlay.

By default the rotation angle is 0.0, it can be any value between 0.0 and 360.0

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation Process with HelpNDoc's Advanced Features](#)

## ImageOverlay\_StretchToVideoSize

### TVideoGrabber.ImageOverlay\_StretchToVideoSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Stretches the current image overlay selected by [ImageOverlaySelector](#) to the video size

#### Declaration

**property** ImageOverlay\_StretchToVideoSize: Boolean **read** GetImageOverlay\_StretchToVideoSize **write** SetImageOverlay\_StretchToVideoSize **default** DEF\_ImageOverlay\_StretchToVideoSize;

**\_\_property bool** ImageOverlay\_StretchToVideoSize=read=GetImageOverlay\_StretchToVideoSize, write=SetImageOverlay\_StretchToVideoSize, **default**=0



Property ImageOverlay\_StretchToVideoSize as Boolean

### Description

When enabled this property stretches the image overlay to the video size, (in this case the image overlay location and size settings are ignored)

See [Graphic overlays](#)

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEONDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

## ImageOverlay\_TopLocation

### TVideoGrabber.ImageOverlay\_TopLocation

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the top location of the current image overlay selected by [ImageOverlaySelector](#)

### Declaration

**property** ImageOverlay\_TopLocation: LongInt **read** GetImageOverlay\_TopLocation **write** SetImageOverlay\_TopLocation **default** DEF\_ImageOverlay\_TopLocation;

**\_\_property** int ImageOverlay\_TopLocation==GetImageOverlay\_TopLocation, write=SetImageOverlay\_TopLocation, **default**=10;

property ImageOverlay\_TopLocation as Long

### Description

Specifies the top location of the overlaid image over the video frames.

The value must be between 0 and the current video height.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## ImageOverlay\_Transparent

### TVideoGrabber.ImageOverlay\_Transparent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the transparency of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_Transparent: Boolean **read** GetImageOverlay\_Transparent **write** SetImageOverlay\_Transparent **default** DEF\_ImageOverlay\_Transparent;

**\_\_property** bool ImageOverlay\_Transparent==GetImageOverlay\_Transparent, write=SetImageOverlay\_Transparent, **default**=0;

property ImageOverlay\_Transparent as Boolean

#### Description

Used to enable the transparency of the image overlay.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## ImageOverlay\_TransparentColorValue

### TVideoGrabber.ImageOverlay\_TransparentColorValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the transparency color of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_TransparentColorValue: LongInt **read** GetImageOverlay\_TransparentColorValue **write** SetImageOverlay\_TransparentColorValue **default** DEF\_ImageOverlay\_TransparentColorValue;

**\_\_property** int ImageOverlay\_TransparentColorValue==GetImageOverlay\_TransparentColorValue, write=SetImageOverlay\_TransparentColorValue, **default**=0;

property ImageOverlay\_TransparentColorValue as Long

#### Description

Used to specify the value of the color used for transparency when [ImageOverlay\\_Transparent](#) and [ImageOverlay\\_UseTransparentColor](#) are enabled.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: Make Help Documentation a Breeze with a Help Authoring Tool

## ImageOverlay\_UseTransparentColor

### TVideoGrabber.ImageOverlay\_UseTransparentColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the use of a transparency color for the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_UseTransparentColor: Boolean **read** GetImageOverlay\_UseTransparentColor **write** SetImageOverlay\_UseTransparentColor **default** DEF\_ImageOverlay\_UseTransparentColor;

**\_\_property** bool ImageOverlay\_UseTransparentColor==GetImageOverlay\_UseTransparentColor, write=SetImageOverlay\_UseTransparentColor, **default**=0;

property ImageOverlay\_UseTransparentColor as Boolean

#### Description

Used to specify if a given color must be used for transparency when [ImageOverlay\\_Transparent](#) is enabled.

If enabled, the color must be specified by [ImageOverlay.TransparentColorValue](#).

If disabled, the default color transparency (background color) of the image will be used.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay.AlphaBlend](#)  
[ImageOverlay.AlphaBlendValue](#) [ImageOverlay.Height](#) [ImageOverlay.LeftLocation](#)  
[ImageOverlay.TopLocation](#) [ImageOverlay.Transparent](#) [ImageOverlay.TransparentColorValue](#)  
[ImageOverlay.VideoAlignment](#) [ImageOverlay.Width](#) [ImageOverlay.Enabled](#) [ImageOverlay.Selector](#)  
[SetImageOverlay.AlphaBlend](#) [SetImageOverlay.AlphaBlendValue](#) [SetImageOverlay.ChromaKey](#)  
[SetImageOverlay.ChromaKeyLeewayPercent](#) [SetImageOverlay.ChromaKeyRGBColor](#)  
[SetImageOverlay.Enabled](#) [SetImageOverlay.Height](#) [SetImageOverlay.LeftLocation](#)  
[SetImageOverlay.RotationAngle](#) [SetImageOverlay.StretchToVideoSize](#) [SetImageOverlay.TargetDisplay](#)  
[SetImageOverlay.TopLocation](#) [SetImageOverlay.Transparent](#) [SetImageOverlay.TransparentColorValue](#)  
[SetImageOverlay.UseTransparentColor](#) [SetImageOverlay.Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Experience the Power and Simplicity of HelpNDoc's User Interface](#)

## ImageOverlay\_VideoAlignment

### TVideoGrabber.ImageOverlay\_VideoAlignment

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Relative alignment of the image overlay within the video frame

#### Declaration

**property** ImageOverlay\_VideoAlignment: TVideoAlignment **read** GetImageOverlayVideoAlignment **write** SetImageOverlayVideoAlignment;

**\_\_property** TVideoAlignment ImageOverlay\_VideoAlignment==GetImageOverlayVideoAlignment, write=SetImageOverlayVideoAlignment, **default**};

#### Description

[TVideoAlignment](#) relative alignment of the image overlay within the video frame.

Default: oa\_LeftTop

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay.AlphaBlend](#)  
[ImageOverlay.AlphaBlendValue](#) [ImageOverlay.Height](#) [ImageOverlay.LeftLocation](#)  
[ImageOverlay.TopLocation](#) [ImageOverlay.Transparent](#) [ImageOverlay.TransparentColorValue](#)  
[ImageOverlay.UseTransparentColor](#) [ImageOverlay.Width](#) [ImageOverlay.Enabled](#) [ImageOverlay.Selector](#)  
[SetImageOverlay.AlphaBlend](#) [SetImageOverlay.AlphaBlendValue](#) [SetImageOverlay.ChromaKey](#)  
[SetImageOverlay.ChromaKeyLeewayPercent](#) [SetImageOverlay.ChromaKeyRGBColor](#)  
[SetImageOverlay.Enabled](#) [SetImageOverlay.Height](#) [SetImageOverlay.LeftLocation](#)  
[SetImageOverlay.RotationAngle](#) [SetImageOverlay.StretchToVideoSize](#) [SetImageOverlay.TargetDisplay](#)  
[SetImageOverlay.TopLocation](#) [SetImageOverlay.Transparent](#) [SetImageOverlay.TransparentColorValue](#)  
[SetImageOverlay.UseTransparentColor](#) [SetImageOverlay.Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

## ImageOverlay\_Width

### TVideoGrabber.ImageOverlay\_Width

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the width of the current image overlay selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlay\_Width: LongInt **read** GetImageOverlay\_Width **write** SetImageOverlay\_Width  
**default** DEF\_ImageOverlay\_Width;

\_\_**property int** ImageOverlay\_Width==GetImageOverlay\_Width, write=SetImageOverlay\_Width, **default=-1**;

property ImageOverlay\_Width as Long

#### Description

Specifies the width to which the overlayed image will be stretched.

Set ImageOverlay\_Width = -1 to use the real width of the loaded image.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#)  
[ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#)  
[ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#)  
[ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlayEnabled](#)  
[ImageOverlaySelector](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#)  
[SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#)  
[SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#)  
[SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#)  
[SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#)  
[SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#)  
[SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Transform your help documentation into a stunning website](#)

## ImageOverlayEnabled

### TVideoGrabber.ImageOverlayEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables / disables the image overlay currently loaded selected by [ImageOverlaySelector](#)

#### Declaration

**property** ImageOverlayEnabled: Boolean **read** GetImageOverlayEnabled **write** SetImageOverlayEnabled  
**default** DEF\_ImageOverlayEnabled;

\_\_**property bool** ImageOverlayEnabled=read=GetImageOverlayEnabled, write=SetImageOverlayEnabled, **default=0**

Property ImageOverlayEnabled As Boolean

## Description

Used to enable/disable the image overlay currently loaded.

*Note: see the [Image overlays](#) chapter for global information about this feature.*

## See Also

[Image overlays](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlaySelector](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

---

## ImageOverlaySelector

### TVideoGrabber.ImageOverlaySelector

[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Select a group of image overlay settings

## Declaration

**property** ImageOverlaySelector: LongInt **read** GetImageOverlaySelector **write** SetImageOverlaySelector;

**\_\_property** int ImageOverlaySelector==GetImageOverlaySelector, write=SetImageOverlaySelector, **nodefault**};

property ImageOverlaySelector as Long

## Description

Used to select a group of image overlay settings.

Up to 1000 image overlay groups of settings may be used, allowing to perform different image overlays (with different location, image format, etc...) at the same time.

The default image overlay settings correspond to the group 0 ([ImageOverlaySelector](#) = 0).

*Note that TVideoGrabber does not allocate the resources for 1000 image overlays, by default it allocates the settings corresponding to ImageOverlaySelector = 0 (the resource allocation is made when ImageOverlaySelector receives a new value never used before, see the [Image overlays](#) chapter for more explanations).*

Set [ImageOverlaySelector](#) to the index of the image overlay properties to read or write before reading or writing their values.

It is applied to all the ImageOverlay... properties and functions;

For example, the following code overlays 2 images at 2 different location, the 1st is overlayed in its current size (width = -1 and height = -1), the second is stretched (because the width and height are specified)



```

procedure TfrmMainForm.Button1Click(Sender: TObject);
begin
    VideoGrabber.ImageOverlaySelector := 0;
    VideoGrabber.SetImageOverlayFromAnyImageFile ('myimage1.gif');
    VideoGrabber.SetImageOverlayAttributes (10, 10, -1, -1, false, false, 0, false, 0);
    VideoGrabber.ImageOverlayEnabled := True;

    VideoGrabber.ImageOverlaySelector := 1;
    VideoGrabber.SetImageOverlayFromAnyImageFile ('myimage2.png');
    VideoGrabber.SetImageOverlayAttributes (100, 100, 30, 20, false, false, 0, false, 0);
    VideoGrabber.ImageOverlayEnabled := True;
end;

```

in C++:

```

void __fastcall TfrmMainForm::Button1Click(TObject *Sender)

void __fastcall TForm1::Button1Click(TObject *Sender)

    VideoGrabber->ImageOverlaySelector = 0;
    VideoGrabber->SetImageOverlayFromAnyImageFile ("myimage1.gif");
    VideoGrabber->SetImageOverlayAttributes (10, 10, -1, -1, false, false, 0, false, 0);
    VideoGrabber->ImageOverlayEnabled = true;

    VideoGrabber->ImageOverlaySelector = 1;
    VideoGrabber->SetImageOverlayFromAnyImageFile ("myimage2.png");
    VideoGrabber->SetImageOverlayAttributes (100, 100, 30, 20, false, false, 0, false, 0);
    VideoGrabber->ImageOverlayEnabled = true;

```

in VB:

```

Private Sub Command1_Click()
    VideoGrabber.ImageOverlaySelector = 0
    VideoGrabber.SetImageOverlayFromAnyImageFile ("myimage1.gif")
    VideoGrabber.SetImageOverlayAttributes (10, 10, -1, -1, false, false, 0, false, 0)
    VideoGrabber.ImageOverlayEnabled = true

    VideoGrabber.ImageOverlaySelector = 1
    VideoGrabber.SetImageOverlayFromAnyImageFile ("myimage2.png")
    VideoGrabber.SetImageOverlayAttributes (100, 100, 30, 20, false, false, 0, false, 0)
    VideoGrabber.ImageOverlayEnabled = true
End Sub

```

### See Also

[Image overlays](#)
[GetPixelsDistance](#)
[GetRGBPixelAt](#)
[ImageOverlay](#)
[AlphaBlend](#)
[ImageOverlay](#)
[AlphaBlendValue](#)
[ImageOverlay](#)
[Height](#)
[ImageOverlay](#)
[LeftLocation](#)
[ImageOverlay](#)
[TopLocation](#)
[ImageOverlay](#)
[Transparent](#)
[ImageOverlay](#)
[TransparentColorValue](#)
[ImageOverlay](#)
[UseTransparentColor](#)
[ImageOverlay](#)
[VideoAlignment](#)
[ImageOverlay](#)
[Width](#)
[ImageOverlayEnabled](#)
[SetImageOverlay](#)
[AlphaBlend](#)
[SetImageOverlay](#)
[AlphaBlendValue](#)
[SetImageOverlay](#)
[ChromaKey](#)
[SetImageOverlay](#)
[ChromaKeyLeewayPercent](#)
[SetImageOverlay](#)
[ChromaKeyRGBColor](#)
[SetImageOverlay](#)
[Enabled](#)
[SetImageOverlay](#)
[Height](#)
[SetImageOverlay](#)
[LeftLocation](#)
[SetImageOverlay](#)
[RotationAngle](#)
[SetImageOverlay](#)
[StretchToVideoSize](#)



[SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

## ImageRatio

### TVideoGrabber.ImageRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Ratio between the width and height of the native video size.

#### Declaration

**property** ImageRatio: Double **read** GetImageRatio;

\_\_property **double** ImageRatio=read=GetImageRatio

Property ImageRatio As Double

#### Description

Used to retrieve the ratio between the width and the height of the native video size.  
E.g. for a 4 / 3 image, this property will return 0.75

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's HTML5 template](#)

## InFrameProgressEvent

### TVideoGrabber.InFrameProgressEvent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the [OnFrameProgress](#) event is currently occurring.

#### Declaration

**property** InFrameProgressEvent: Boolean **read** GetInFrameProgressEvent;

\_\_property **bool** InFrameProgressEvent=read=GetInFrameProgressEvent, **nodefault**

Property InFrameProgressEvent As Boolean

#### Description

Used to know if the [OnFrameProgress](#) event is currently occurring.

This property is useful e.g. when using a trackbar to track the player position. In this case the TrackBar's OnChange event must not update the TVideoGrabber's player position if this event occurs while the

trackbar is updated from the OnFrameProgress event, otherwise a loop back update occurs and the trackbar motion is "jerky".

Sample code:

```
procedure TForm1.TrackBar1Change(Sender: TObject);
begin
    if not VideoGrabber1.InFrameProgressEvent then begin
        VideoGrabber1.PlayerFramePosition := TrackBar1.Position;
    end;
end;
```

**See Also**

[TFrameGrabberRGBFormat](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#) [FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#) [GetFrameInfoString](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## IPCameraURL

### TVideoGrabber.IPCameraURL

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets the URL of the IP Camera

**Declaration**

**property** IPCameraURL: **String** **read** GetIPCameraURL **write** SetIPCameraURL

**\_\_property** wchar\_t \*IPCameraURL==GetIPCameraURL, write=SetIPCameraURL

Property IPCameraURL as String

**Description**

Used to specify the URL of the IP Camera when [VideoSource](#) = vs\_IPCamera

E.g. for an Axis camera:

(x.x.x.x is the IP of the camera)

VideoGrabber.VideoSource = vs\_IPCamera

VideoGrabber.IPCameraURL = "http://x.x.x.x/axis-cgi/mjpg/video.cgi?resolution=640x480

VideoGrabber.StartPreview

**See Also**

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF](#) [GetStr](#) [ONVIF](#) [SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's User-Friendly UI](#)

## IsAnalogVideoDecoderAvailable

### TVideoGrabber.IsAnalogVideoDecoderAvailable

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Returns true if an [analog video decoder](#) is available.

### Declaration

**property** IsAnalogVideoDecoderAvailable: Boolean **read** GetIsAnalogVideoDecoderAvailable;

\_\_property **bool** IsAnalogVideoDecoderAvailable=read=GetIsAnalogVideoDecoderAvailable, **nodefault**

Property IsAnalogVideoDecoderAvailable As Boolean

### Description

Returns true if the current [video capture device](#) exposes an [analog video decoder](#) .

### See Also

[Analog Video standards](#) [AnalogVideoStandard](#) [AnalogVideoStandardIndex](#) [AnalogVideoStandards](#) [AnalogVideoStandardsCount](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

## IsAudioCrossbarAvailable

### TVideoGrabber.IsAudioCrossbarAvailable

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Returns true if the audio crossbar is available.

### Declaration

**property** IsAudioCrossbarAvailable: Boolean **read** GetIsAudioCrossbarAvailable;

\_\_property **bool** IsAudioCrossbarAvailable=read=GetIsAudioCrossbarAvailable, **nodefault**

Property IsAudioCrossbarAvailable As Boolean

### Description

Returns true if the current audio capture device exposes an audio crossbar.

### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation with HelpNDoc's Project Analyzer Features](#)

## IsAudioInputBalanceAvailable

### TVideoGrabber.IsAudioInputBalanceAvailable

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Availability of the audio input balance.

### Declaration

**property** IsAudioInputBalanceAvailable: TTriState **read** GetIsAudioInputBalanceAvailable;

\_\_property TTriState IsAudioInputBalanceAvailable=read=GetIsAudioInputBalanceAvailable, **nodefault**

Property IsAudioInputBalanceAvailable As TTriState

### Description

This property lets you know if audio balance is available for the current [audio input](#) selected.

This is a tri-state property, that returns ts\_True if available, or ts\_False if not available.

The [preview](#) or [recording](#) must be running, otherwise this property returns ts\_Undefined.

### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## IsCameraControlAvailable

### TVideoGrabber.IsCameraControlAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the camera controls are available.

### Declaration

**property** IsCameraControlAvailable: Boolean **read** GetIsCameraControlAvailable;

\_\_property **bool** IsCameraControlAvailable=read=GetIsCameraControlAvailable, **nodefault**

Property IsCameraControlAvailable As Boolean

### Description

Returns true if the current video capture device exposes camera control settings.

### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

## IsDigitalVideoIn

### TVideoGrabber.IsDigitalVideoIn

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the current video capture device is a [DV](#) source.

### Declaration

**property** IsDigitalVideoIn: Boolean **read** GetIsDigitalVideoIn;

\_\_property **bool** IsDigitalVideoIn=read=GetIsDigitalVideoIn, **nodefault**

Property IsDigitalVideoIn As Boolean

## Description

Returns true if the current [video capture device](#) is a [DV](#) source.

## See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## IsDVCommandAvailable

### TVideoGrabber.IsDVCommandAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if [DV commands](#) are available.

## Declaration

**property** IsDVCommandAvailable: Boolean **read** GetIsDVCommandAvailable;

\_\_property **bool** IsDVCommandAvailable=read=GetIsDVCommandAvailable, **nodefault**

Property IsDVCommandAvailable As Boolean

## Description

Returns true if [DV commands](#) are available on the current video capture device.

## See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## IsHorizontalSyncLocked

### TVideoGrabber.IsHorizontalSyncLocked

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true upon locked horizontal sync.

## Declaration

**property** IsHorizontalSyncLocked: Boolean **read** GetIsHorizontalSyncLocked;

\_\_property **bool** IsHorizontalSyncLocked=read=GetIsHorizontalSyncLocked, **nodefault**

Property IsHorizontalSyncLocked As Boolean

## Description

Returns true if the horizontal synchronization of the analog video decoder is locked.

## See Also

[TDVCommand](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [SendDVCommand](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## IsMPEGStream

### TVideoGrabber.IsMPEGStream

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Returns true if the video capture devices outputs an MPEG stream.

#### Declaration

**property** IsMpegStream: Boolean **read** GetIsMpegStream;

\_\_property **bool** IsMpegStream=read=GetIsMpegStream, **nodefault**

Property IsMpegStream As Boolean

#### Description

Returns true if the video capture devices outputs an MPEG stream.

Created with the Standard Edition of HelpNDoc: [Experience the Power and Simplicity of HelpNDoc's User Interface](#)

## IsPlayerAudioStreamAvailable

### TVideoGrabber.IsPlayerAudioStreamAvailable

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Availability of the audio stream in a clip.

#### Declaration

**property** IsPlayerAudioStreamAvailable: TTriState **read** GetIsPlayerAudioStreamAvailable;

\_\_property TTriState IsPlayerAudioStreamAvailable=read=GetIsPlayerAudioStreamAvailable, **nodefault**

Property IsPlayerAudioStreamAvailable As TTriState

#### Description

This property lets you know if a clip includes an audio stream.

This is a tri-state property, that returns **ts\_True** if an audio stream is available, or **ts\_False** if there is no audio in the clip.

The clip must have been opened with [OpenPlayer](#), otherwise this property returns **ts\_Undefined**.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## IsPlayerVideoStreamAvailable

### TVideoGrabber.IsPlayerVideoStreamAvailable

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Availability of the video stream in a clip.

#### Declaration

**property** IsPlayerVideoStreamAvailable: TTriState **read** GetIsPlayerVideoStreamAvailable;

\_\_property TTriState IsPlayerVideoStreamAvailable=read=GetIsPlayerVideoStreamAvailable, **nodefault**

Property IsPlayerVideoStreamAvailable As TTriState

#### Description

This property lets you know if a clip includes a video stream.

This is a tri-state property, that returns **ts\_True** if a video stream is available, or **ts\_False** if there no video stream.

The clip must have been opened with [OpenPlayer](#), otherwise this property returns **ts\_Undefined**.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## IsRecordingPaused

### TVideoGrabber.IsRecordingPaused

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the current recording is in a paused state.

#### Declaration

**property** IsRecordingPaused: Boolean **read** GetIsRecordingPaused;

\_\_property **bool** IsRecordingPaused=read=GetIsRecordingPaused, **nodefault**

Property IsRecordingPaused As Boolean

#### Description

Used to retrieve whether the current recording has been paused by [PauseRecording](#) .

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#)



[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## IsTimeCodeReaderAvailable

### TVideoGrabber.IsTimeCodeReaderAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#)
[Properties](#)

Returns true if time code information is available.

#### Declaration

**property** IsTimeCodeReaderAvailable: Boolean **read** GetTimeCodeReaderAvailable;

\_\_property **bool** IsTimeCodeReaderAvailable=read=GetTimeCodeReaderAvailable, **nodefault**

Property IsTimeCodeReaderAvailable As Boolean

#### Description

Returns true if time code information is available for the current [video capture device](#) .

#### See Also

[TDVSize](#)
[TDVVideoFormat](#)
[TDVVideoStandard](#)
[TOnDVCommandCompleted](#)
[DVDDateTimeEnabled](#)
[DVDDiscontinuityMinimumInterval](#)
[DVRRecordingInNativeFormatSeparatesStreams](#)
[DVReduceFrameRate](#)
[DVRgb219](#)
[DVTimeCodeEnabled](#)
[IsDigitalVideoIn](#)
[IsDVCommandAvailable](#)
[OnDVCommandCompleted](#)
[SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

## IsTVAudioAvailable

### TVideoGrabber.IsTVAudioAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#)
[Properties](#)

Returns true if the TV audio is available.

#### Declaration

**property** IsTVAudioAvailable: Boolean **read** GetIsTVAudioAvailable;

\_\_property **bool** IsTVAudioAvailable=read=GetIsTVAudioAvailable, **nodefault**

Property IsTVAudioAvailable As Boolean

#### Description

Returns true if the TV audio filter is available for the current video capture device.

#### See Also

[TTVChannelInfo](#)
[TOnTVChannelScanStarted](#)
[TOnTVChannelSelected](#)
[TTunerInputType](#)
[GetTVChannelInfo](#)
[IsHorizontalSyncLocked](#)
[IsTVAutoTuneRunning](#)
[IsTVTunerAvailable](#)
[OnTVChannelScanCompleted](#)

[OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#)  
[TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#)  
[TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## IsTVAutoTuneRunning

### TVideoGrabber.IsTVAutoTuneRunning

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true when the [TV tuner](#) automatic tuning is currently running.

#### Declaration

**property** IsTVAutoTuneRunning: Boolean **read** GetIsTVAutoTuneRunning;

\_\_property **bool** IsTVAutoTuneRunning=read=GetIsTVAutoTuneRunning, **nodefault**

Property IsTVAutoTuneRunning As Boolean

#### Description

Used to retrieve whether the [TV tuner](#) automatic tuning is currently running.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#)  
[IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#)  
[OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#)  
[TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#)  
[TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Elevate your documentation to new heights with HelpNDoc's built-in SEO](#)

## IsTVTunerAvailable

### TVideoGrabber.IsTVTunerAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the [TV tuner](#) is available.

#### Declaration

**property** IsTVTunerAvailable: Boolean **read** GetIsTVTunerAvailable;

\_\_property **bool** IsTVTunerAvailable=read=GetIsTVTunerAvailable, **nodefault**

Property IsTVTunerAvailable As Boolean

#### Description

Returns true if the [TV tuner](#) is available for the current video capture device.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#)  
[IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [OnTVChannelScanCompleted](#)  
[OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#)  
[TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#)  
[TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## IsVideoControlAvailable

### TVideoGrabber.IsVideoControlAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if [video control modes](#) are available.

#### Declaration

**property** IsVideoControlAvailable: Boolean **read** GetIsVideoControlAvailable;

\_\_property **bool** IsVideoControlAvailable=read=GetIsVideoControlAvailable, **nodefault**

Property IsVideoControlAvailable As Boolean

#### Description

Returns true if [video control modes](#) can be set on the current video capture device.

#### See Also

[TVideoControl](#) [GetVideoControlMode](#) [IsVideoControlModeAvailable](#) [SetVideoControlMode](#) [SetVideoControlMode2](#) [VideoControlSettings](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

## IsVideoCrossbarAvailable

### TVideoGrabber.IsVideoCrossbarAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if a video crossbar is available.

#### Declaration

**property** IsVideoCrossbarAvailable: Boolean **read** GetIsVideoCrossbarAvailable;

\_\_property **bool** IsVideoCrossbarAvailable=read=GetIsVideoCrossbarAvailable, **nodefault**

Property IsVideoCrossbarAvailable As Boolean

#### Description

Returns true if a video crossbar is available for the current video capture device.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

## IsVideoInterlaced

### TVideoGrabber.IsVideoInterlaced

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the current video capture device source is interlaced.

**Declaration**

**property** IsVideoInterlaced: Boolean **read** GetIsVideoInterlaced;

\_\_\_property **bool** IsVideoInterlaced=read=GetIsVideoInterlaced, **nodefault**

Property IsVideoInterlaced As Boolean

**Description**

Returns true if the current [video capture device](#) video source is interlaced.

**See Also**

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

**IsVideoPortAvailable****TVideoGrabber.IsVideoPortAvailable**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the video capture device has a video port.

**Declaration**

**property** IsVideoPortAvailable: Boolean **read** GetIsVideoPortAvailable;

\_\_\_property **bool** IsVideoPortAvailable=read=GetIsVideoPortAvailable, **nodefault**

Property IsVideoPortAvailable As Boolean

**Description**

Used to know if the video capture device has a video port.

See [Display\\_VideoPortEnabled](#) .

**See Also**

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

## IsVideoQualityAvailable

### TVideoGrabber.IsVideoQualityAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if [video quality settings](#) are available.

#### Declaration

**property** IsVideoQualityAvailable: Boolean **read** GetIsVideoQualityAvailable;

\_\_property **bool** IsVideoQualityAvailable=read=GetIsVideoQualityAvailable, **nodefault**

Property IsVideoQualityAvailable As Boolean

#### Description

Returns true if [video quality settings](#) are available on the current video capture device.

#### See Also

[IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#) [VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Easy EBook and documentation generator](#)

## IsWDMVideoDriver

### TVideoGrabber.IsWDMVideoDriver

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the video capture driver is WDM compliant.

#### Declaration

**property** IsWDMVideoDriver: Boolean **read** GetIsWDMVideoDriver;

\_\_property **bool** IsWDMVideoDriver=read=GetIsWDMVideoDriver, **nodefault**

Property IsWDMVideoDriver As Boolean

#### Description

Returns:

- true if the current video capture device driver is a WDM driver,
- false if the current video capture device driver is an "old" VFW driver.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)

## JPEGPerformance

### TVideoGrabber.JPEGPerformance

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Controls the trade-off between color quality and speed of decompression of JPEG files created by the frame grabber.

#### Declaration

**property** JPEGPerformance: TJPEGPerformance **read** GetJPEGPerformance **write** SetJPEGPerformance **default** DEF\_JPEGPerformance;

\_\_property Jpeg::TJPEGPerformance JPEGPerformance=read=GetJPEGPerformance, write=SetJPEGPerformance, **default**=0

Property JPEGPerformance As TxJPEGPerformance

#### Description

Use JPEGPerformance to control the trade-off between color quality and speed of decompression of JPEG files created during [frame capture](#).

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## JPEGProgressiveDisplay

### TVideoGrabber.JPEGProgressiveDisplay

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Turns on or off the incremental display of an image when reading in JPEG files created by the frame grabber.

#### Declaration

**property** JPEGProgressiveDisplay: Boolean **read** GetJPEGProgressiveDisplay **write** SetJPEGProgressiveDisplay **default** DEF\_JPEGProgressiveDisplay;

\_\_property **bool** JPEGProgressiveDisplay=read=GetJPEGProgressiveDisplay, write=SetJPEGProgressiveDisplay, **default**=0

Property JPEGProgressiveDisplay As Boolean

#### Description

Use ProgressiveDisplay to turn on or off the incremental display of an image when reading in JPEG files created during [frame capture](#).

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

## JPEGQuality

### TVideoGrabber.JPEGQuality

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets the compression quality of a JPEG image created by the frame grabber.

#### Declaration

**property** JPEGQuality: TJPEGQualityRange **read** GetJPEGQuality **write** SetJPEGQuality **default** DEF\_JPEGQuality;

\_\_property Jpeg::TJPEGQualityRange JPEGQuality=read=GetJPEGQuality, write=SetJPEGQuality, default=100

Property JPEGQuality As Integer

#### Description

Use CompressionQuality to set the compression quality of a JPEG image during [frame capture](#). Higher compression results in a poorer picture quality, but a smaller file size.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## Last\_BurstFrameCapture\_FileName

### TVideoGrabber.Last\_BurstFrameCapture\_FileName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Name of the last BMP or JPEG file created by the frame capture in burst mode

#### Declaration

**property** Last\_BurstFrameCapture\_FileName: **string** **read** GetLast\_BurstFrameCapture\_FileName;

\_\_property wchar\_t \*Last\_BurstFrameCapture\_FileName=read=GetLast\_BurstFrameCapture\_FileName

Property Last\_BurstFrameCapture\_FileName As String

#### Description

Retrieves the name of the last BMP or JPEG file created automatically by the frame grabber in burst mode. This property should be invoked from the [OnFrameCaptureCompleted](#) event.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)



## Last\_CaptureFrameTo\_FileName

### TVideoGrabber.Last\_CaptureFrameTo\_FileName

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Name of the last BMP or JPEG file created by CaptureFrameTo.

#### Declaration

**property** Last\_CaptureFrameTo\_FileName: **string read** GetLast\_CaptureFrameTo\_FileName;

\_\_property wchar\_t \*Last\_CaptureFrameTo\_FileName=read=GetLast\_CaptureFrameTo\_FileName

Property Last\_CaptureFrameTo\_FileName As String

#### Description

Retrieves the name of the last BMP or JPEG file created automatically by [CaptureFrameTo](#). This property should be invoked from the [OnFrameCaptureCompleted](#) event.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last\\_BurstFrameCapture](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

## Last\_Clip\_Played

### TVideoGrabber.Last\_Clip\_Played

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Path/file name of the last video clip played.

#### Declaration

**property** Last\_Clip\_Played: **string read** GetLast\_Clip\_Played;

\_\_property wchar\_t \*Last\_Clip\_Played=read=GetLast\_Clip\_Played

Property Last\_Clip\_Played As String

#### Description

Retrieves the path/file name of the last video clip played.

E.g.:

```
procedure TfrmMainForm.Button1Click(Sender: TObject);
begin
    VideoGrabber.PlayerFileName := VideoGrabber.Last_Clip_Played;
    VideoGrabber.OpenPlayer;
end;
```

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Transform Your Documentation Process with HelpNDoc's Project Analyzer

## Last\_Recording\_FileName

### TVideoGrabber.Last\_Recording\_FileName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Name of the last AVI file created

#### Declaration

**property** Last\_Recording\_FileName: **string read** GetLast\_Recording\_FileName;

\_\_property wchar\_t \*Last\_Recording\_FileName=read=GetLast\_Recording\_FileName

Property Last\_Recording\_FileName As String

#### Description

Retrieves the name of the last AVI file created (useful when file name are created automatically, see [recording](#) ).

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Create cross-platform Qt Help files

## LogoDisplayed

### TVideoGrabber.LogoDisplayed

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables / disables the video window logo.

**Declaration**

**property** LogoDisplayed: Boolean **read** GetLogoDisplayed **write** SetLogoDisplayed **default** DEF\_LogoDisplayed;

\_\_property **bool** LogoDisplayed=read=GetLogoDisplayed, write=SetLogoDisplayed, **default**=0

Property LogoDisplayed As Boolean

**Description**

Used to enable or disable the logo displayed in the inactive video window.

**See Also**

[Logo displayed in the video window](#) [LogoLayout](#) [SetLogoFromBMPFile](#) [SetLogoFromHBitmap](#) [SetLogoFromJPEGFile](#) [SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

**LogoLayout****TVideoGrabber.LogoLayout**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies how the logo must be displayed.

**Declaration**

**property** LogoLayout: TLogoLayout **read** GetLogoLayout **write** SetLogoLayout **default** DEF\_LogoLayout;

\_\_property TLogoLayout LogoLayout=read=GetLogoLayout, write=SetLogoLayout, **default**=1

Property LogoLayout As TxLogoLayout

**Description**

This [TLogoLayout](#) property is used to specify if the logo must be:

- centered (lg\_Centered)
- stretched to fit the video window (lg\_Stretched)
- repeated to fit the video window (lg\_Repeated).

**See Also**

[Logo displayed in the video window](#) [LogoDisplayed](#) [SetLogoFromBMPFile](#) [SetLogoFromHBitmap](#) [SetLogoFromJPEGFile](#) [SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

**MixAudioSamples\_CurrentSourceLevel****TVideoGrabber.MixAudioSamples\_CurrentSourceLevel**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

specifies the level of the current source

**Declaration**

**property** MixAudioSamples\_CurrentSourceLevel: LongInt **index** AUDMIX\_CURRENT **read** GetMixAudioSamplesLevel **write** SetMixAudioSamplesLevel **default** 100;

\_\_property **int** MixAudioSamples\_CurrentSourceLevel==GetMixAudioSamplesLevel,

write=SetMixAudioSamplesLevel, index=0, **default**=100;

property MixAudioSamples\_CurrentSourceLevel as Long

#### Description

Used to specify the level of the current source, in percentage, when mixing audio samples by using [MixAudioSamples](#).

#### See Also

[MixAudioSamples](#) [MixAudioSamples\\_ExternalSourceLevel](#) [Mixer\\_SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

### MixAudioSamples\_ExternalSourceLevel

#### TVideoGrabber.MixAudioSamples\_ExternalSourceLevel

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

specifies the level of the external source

#### Declaration

**property** MixAudioSamples\_ExternalSourceLevel: LongInt **index** AUDMIX\_EXTERN **read** GetMixAudioSamplesLevel **write** SetMixAudioSamplesLevel **default** DEF\_MixAudioSamplesLevel;

**\_\_property** int MixAudioSamples\_ExternalSourceLevel==GetMixAudioSamplesLevel, write=SetMixAudioSamplesLevel, index=1, **default**=100;

property MixAudioSamples\_ExternalSourceLevel as Long

#### Description

Used to specify the level of the external source, in percentage, when mixing audio samples by using [MixAudioSamples](#).

#### See Also

[MixAudioSamples](#) [MixAudioSamples\\_CurrentSourceLevel](#) [Mixer\\_SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

### Mixer\_MosaicColumns

#### TVideoGrabber.Mixer\_MosaicColumns

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the number of mosaic columns in Mixer mode.

#### Declaration

**property** Mixer\_MosaicColumns: LongInt **read** GetMixer\_MosaicColumns **write** SetMixer\_MosaicColumns **default** DEF\_Mixer\_MosaicColumns;

**\_\_property** int Mixer\_MosaicColumns==GetMixer\_MosaicColumns, write=SetMixer\_MosaicColumns, **default**=1;

Property Mixer\_MosaicColumns as Long

#### Description

Used to specify the number of columns when the component is used to mix several video sources into a

single one in a mosaic layout (see [How to mix several video sources into one a single one](#)).

#### See Also

[Mixer Activation](#) [Mixer AddToMixer](#) [Mixer MosaicLines](#) [Mixer RemoveFromMixer](#)  
[Mixer SetOverlayRoundedCorner](#) [Mixer SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## Mixer\_MosaicLines

### TVideoGrabber.Mixer\_MosaicLines

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the number of mosaic lines in Mixer mode.

#### Declaration

**property** Mixer\_MosaicLines: LongInt **read** GetMixer\_MosaicLines **write** SetMixer\_MosaicLines **default** DEF\_Mixer\_MosaicLines;

\_\_**property int** Mixer\_MosaicLines==GetMixer\_MosaicLines, write=SetMixer\_MosaicLines, **default**=1;

Property Mixer\_MosaicLines as Long

#### Description

Used to specify the number of lines when the component is used to mix several video sources into a single one in a mosaic layout (see [How to mix several video sources into one a single one](#)).

#### See Also

[Mixer Activation](#) [Mixer AddToMixer](#) [Mixer MosaicColumns](#) [Mixer RemoveFromMixer](#)  
[Mixer SetOverlayRoundedCorner](#) [Mixer SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Protect Your Confidential PDFs with These Simple Security Measures](#)

## MotionDetector\_CompareBlue

### TVideoGrabber.MotionDetector\_CompareBlue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the use of the blue color to detect motion.

#### Declaration

**property** MotionDetector\_CompareBlue: Boolean **read** GetMotionDetector\_CompareBlue **write** SetMotionDetector\_CompareBlue **default** DEF\_MotionDetector\_CompareBlue;

\_\_**property bool** MotionDetector\_CompareBlue=read=GetMotionDetector\_CompareBlue, write=SetMotionDetector\_CompareBlue, **default**=1

Property MotionDetector\_CompareBlue As Boolean

#### Description

Used to enable/disable the use of the blue color to detect motion.  
 If [MotionDetector\\_GreyScale](#) is enabled this property is ignored.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#)  
[Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#)  
[MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#)

[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)
[MotionDetector\\_GetCellSize](#)
[MotionDetector\\_GloballyIncOrDecSensitivity](#)
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)
[MotionDetector\\_Grid](#)
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## MotionDetector\_CompareGreen

### TVideoGrabber.MotionDetector\_CompareGreen

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the use of the green color to detect motion.

#### Declaration

**property** MotionDetector\_CompareGreen: Boolean **read** GetMotionDetector\_CompareGreen **write** SetMotionDetector\_CompareGreen **default** DEF\_MotionDetector\_CompareGreen;

\_\_property **bool** MotionDetector\_CompareGreen=read=GetMotionDetector\_CompareGreen, write=SetMotionDetector\_CompareGreen, **default**=1

Property MotionDetector\_CompareGreen As Boolean

#### Description

Used to enable/disable the use of the green color to detect motion.  
If [MotionDetector\\_GreyScale](#) is enabled this property is ignored.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#)
[Motion ratio](#)
[Recording only when motion is detected](#)
[Video noise](#)
[TOnMotionDetected](#)
[TOnMotionNotDetected](#)
[MotionDetector\\_CellMotionRatio](#)
[MotionDetector\\_CompareBlue](#)
[MotionDetector\\_CompareRed](#)
[MotionDetector\\_Enabled](#)
[MotionDetector\\_EnumGridDialogControls](#)
[MotionDetector\\_Get2DTextGrid](#)
[MotionDetector\\_Get2DTextMotion](#)
[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)
[MotionDetector\\_GetCellSize](#)
[MotionDetector\\_GloballyIncOrDecSensitivity](#)
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)
[MotionDetector\\_Grid](#)
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## MotionDetector\_CompareRed

### TVideoGrabber.MotionDetector\_CompareRed

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the use of the red color to detect motion.

#### Declaration

**property** MotionDetector\_CompareRed: Boolean **read** GetMotionDetector\_CompareRed **write**

SetMotionDetector\_CompareRed **default** DEF\_MotionDetector\_CompareRed;

\_\_property **bool** MotionDetector\_CompareRed=read=GetMotionDetector\_CompareRed,  
write=SetMotionDetector\_CompareRed, **default**=1

Property MotionDetector\_CompareRed As Boolean

### Description

Used to enable/disable the use of the red color to detect motion.  
If [MotionDetector\\_GreyScale](#) is enabled this property is ignored.

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected](#)  
[Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_Enabled](#)  
[MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion](#)  
[MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize](#)  
[MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale](#)  
[MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered MotionDetector\\_UseThisReferenceSample OnBacktimedFramesCountReached](#)  
[OnMotionDetected OnMotionNotDetected RecordingOnMotion\\_Enabled](#)  
[RecordingOnMotion MotionThreshold RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Powerful and User-Friendly Help Authoring Tool for  
Markdown Documents

## MotionDetector\_Enabled

### TVideoGrabber.MotionDetector\_Enabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables / disables motion detection.

### Declaration

**property** MotionDetector\_Enabled: Boolean **read** GetMotionDetector\_Enabled **write**  
SetMotionDetector\_Enabled **default** DEF\_MotionDetector\_Enabled;

\_\_property **bool** MotionDetector\_Enabled=read=GetMotionDetector\_Enabled,  
write=SetMotionDetector\_Enabled, **default**=0

Property MotionDetector\_Enabled As Boolean

### Description

Used to enable / disable motion detection.  
 - when disabled, all the motion detection features are not used.  
 - when enabled, motion detection features are in use and the [OnMotionDetected](#) event occurs upon motion detection.

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected](#)  
[Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed](#)  
[MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion](#)  
[MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize](#)  
[MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale](#)  
[MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad](#)



[MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## MotionDetector\_GlobalMotionRatio

### TVideoGrabber.MotionDetector\_GlobalMotionRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the last motion ratio.

#### Declaration

**property** MotionDetector\_GlobalMotionRatio: Double **read** GetMotionDetector\_GlobalMotionRatio;

\_\_property **double** MotionDetector\_GlobalMotionRatio=read=GetMotionDetector\_GlobalMotionRatio

Function MotionDetector\_CellMotionRatio(x As Long, y As Long) As Double

#### Description

Used to retrieve the last motion ratio (moved pixels count / total pixels count) of the grid's cells having their sensitivity enabled (value between 1 and 9).

The return value will be in the **0.0** to **1.0** range (e.g. 0.5775).

This value is the result of the motion detection of the last video frame received.

This value is also returned by the [OnMotionDetected](#) event.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## MotionDetector\_GreyScale

### TVideoGrabber.MotionDetector\_GreyScale

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables / disables motion detection on a greyscale basis.

#### Declaration

**property** MotionDetector\_GreyScale: Boolean **read** GetMotionDetector\_GreyScale **write** SetMotionDetector\_GreyScale **default** DEF\_MotionDetector\_GreyScale;

\_\_property **bool** MotionDetector\_GreyScale=read=GetMotionDetector\_GreyScale, write=SetMotionDetector\_GreyScale, **default**=0

Property MotionDetector\_GreyScale As Boolean

### Description

If enabled, motion detection is performed on a greyscale basis, instead of the default RGB basis. In this case, the [MotionDetector\\_CompareRed](#), [MotionDetector\\_CompareBlue](#) and [MotionDetector\\_CompareGreen](#) properties are ignored.

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion](#) [MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Easy to use tool to create HTML Help files and Help web sites

## MotionDetector\_Grid

### TVideoGrabber.MotionDetector\_Grid

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the grid structure, expressed as a simple string.

### Declaration

**property** MotionDetector\_Grid: **string** **read** GetMotionDetector\_Grid **write** SetMotionDetector\_Grid;

\_\_property wchar\_t \*MotionDetector\_Grid=read=GetMotionDetector\_Grid, write=SetMotionDetector\_Grid

Property MotionDetector\_Grid As String

### Description

Used to set or retrieve the sensitivity grid structure, expressed as a simple string. After modifying the grid structure, test [MotionDetector\\_IsGridValid](#) to check if the grid has been accepted.

By example, the sensitivity grid:

```
0 0 0 0 4 4 4 4 1
5 5 5 5 5 5 5 5 1
4 4 4 4 4 4 4 4 1
5 5 5 5 5 5 5 5 1
6 6 6 6 6 6 6 6 1
6 6 6 6 6 6 6 6 1
6 6 6 6 6 6 6 6 1
3 3 3 3 3 3 3 3 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

has for MotionDetector\_Grid value:

```
"0000044441 5555555551 4444444441 5555555551 6666666661 6666666661 6666666661 3333333331 0000000000 0000000000"
```

### See Also

[Color / Greyscale](#)
[Grid structure / grid sensitivity](#)
[Motion ratio](#)
[Recording only when motion is detected](#)
[Video noise](#)
[TOnMotionDetected](#)
[TOnMotionNotDetected](#)
[MotionDetector\\_CellMotionRatio](#)
[MotionDetector\\_CompareBlue](#)
[MotionDetector\\_CompareGreen](#)
[MotionDetector\\_CompareRed](#)
[MotionDetector\\_Enabled](#)
[MotionDetector\\_EnumGridDialogControls](#)
[MotionDetector\\_Get2DTextGrid](#)
[MotionDetector\\_Get2DTextMotion](#)
[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)
[MotionDetector\\_GetCellSize](#)
[MotionDetector\\_GloballyIncOrDecSensitivity](#)
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Create HTML Help, DOC, PDF and print manuals from 1 single source

## MotionDetector\_GridXCount

### TVideoGrabber.MotionDetector\_GridXCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the number of columns of the grid.

#### Declaration

**property** MotionDetector\_GridXCount: LongInt **read** GetMotionDetector\_GridXCount;

\_\_property **int** MotionDetector\_GridXCount=read=GetMotionDetector\_GridXCount, **nodefault**

Property MotionDetector\_GridXCount As Long

#### Description

Used to retrieve the number of columns of the grid.

#### See Also

[Color / Greyscale](#)
[Grid structure / grid sensitivity](#)
[Motion ratio](#)
[Recording only when motion is detected](#)
[Video noise](#)
[TOnMotionDetected](#)
[TOnMotionNotDetected](#)
[MotionDetector\\_CellMotionRatio](#)
[MotionDetector\\_CompareBlue](#)
[MotionDetector\\_CompareGreen](#)
[MotionDetector\\_CompareRed](#)
[MotionDetector\\_Enabled](#)
[MotionDetector\\_EnumGridDialogControls](#)
[MotionDetector\\_Get2DTextGrid](#)
[MotionDetector\\_Get2DTextMotion](#)
[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)
[MotionDetector\\_GetCellSize](#)
[MotionDetector\\_GloballyIncOrDecSensitivity](#)
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)
[MotionDetector\\_Grid](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Create cross-platform Qt Help files

## MotionDetector\_GridYCount

### TVideoGrabber.MotionDetector\_GridYCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the number of rows of the grid.

#### Declaration

**property** MotionDetector\_GridYCount: LongInt **read** GetMotionDetector\_GridYCount;

\_\_\_property **int** MotionDetector\_GridYCount=read=GetMotionDetector\_GridYCount, **nodefault**

Property MotionDetector\_GridYCount As Long

### Description

Used to retrieve the number of rows of the grid.

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## MotionDetector\_IsGridValid

### TVideoGrabber.MotionDetector\_IsGridValid

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the grid string is correct, false otherwise.

### Declaration

**property** MotionDetector\_IsGridValid: Boolean **read** GetMotionDetector\_IsGridValid;

\_\_\_property **bool** MotionDetector\_IsGridValid=read=GetMotionDetector\_IsGridValid, **nodefault**

Property MotionDetector\_IsGridValid As Boolean

### Description

Returns true if the grid string is correct, false otherwise.

See [Grid structure](#) .

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Add an Extra Layer of Security to Your PDFs with](#)

## Encryption

**MotionDetector\_MaxDetectionsPerSecond****TVideoGrabber.MotionDetector\_MaxDetectionsPerSecond**[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Maximum motion detections per second.

**Declaration**

**property** MotionDetector\_MaxDetectionsPerSecond: Double **read** GetMotionDetector\_MaxDetectionsPerSecond **write** SetMotionDetector\_MaxDetectionsPerSecond;

\_\_property **double**

MotionDetector\_MaxDetectionsPerSecond=read=GetMotionDetector\_MaxDetectionsPerSecond, write=SetMotionDetector\_MaxDetectionsPerSecond

Property MotionDetector\_MaxDetectionsPerSecond As Double

**Description**

Used to specify the maximum number of motion detections per second.

0 = no limit (a motion detection is performed on each video frame).

E.g. 5 = 5 detections per second.

**See Also**

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected](#)

[Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio](#)

[MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed](#)

[MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid](#)

[MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity](#)

[MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity](#)

[MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid](#)

[MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid](#)

[MotionDetector\\_ReduceCPULoad MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset](#)

[MotionDetector\\_ResetGlobalSensitivity MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize](#)

[MotionDetector\\_ShowGridDialog MotionDetector\\_Triggered MotionDetector\\_UseThisReferenceSample](#)

[OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected RecordingOnMotion\\_Enabled](#)

[RecordingOnMotion MotionThreshold RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

**MotionDetector\_ReduceCPULoad****TVideoGrabber.MotionDetector\_ReduceCPULoad**[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Reduces the CPU load when detecting motion on large video frames.

**Declaration**

**property** MotionDetector\_ReduceCPULoad: LongInt **read** GetMotionDetector\_ReduceCPULoad **write** SetMotionDetector\_ReduceCPULoad **default** DEF\_MotionDetector\_ReduceCPULoad;

\_\_property **int** MotionDetector\_ReduceCPULoad=read=GetMotionDetector\_ReduceCPULoad, write=SetMotionDetector\_ReduceCPULoad, **default**=1

Property MotionDetector\_ReduceCPULoad As Long

## Description

Used to reduces the CPU load when detecting motion on large video frames.  
Allowed values:

### 1: full processing, maximum accuracy

full motion detection processing is applied, whatever the video frame size.

### 2 to 9: "lighter" processing

Higher the value, "lighter" the processing applied to detect motion, but lower is the motion detection accuracy.

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool

## MotionDetector\_ReduceVideoNoise

### TVideoGrabber.MotionDetector\_ReduceVideoNoise

[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Reduces the sensitivity to the video noise.

### Declaration

**property** MotionDetector\_ReduceVideoNoise: Boolean **read** GetMotionDetector\_ReduceVideoNoise **write** SetMotionDetector\_ReduceVideoNoise **default** DEF\_MotionDetector\_ReduceVideoNoise;

\_\_property **bool** MotionDetector\_ReduceVideoNoise=read=GetMotionDetector\_ReduceVideoNoise, write=SetMotionDetector\_ReduceVideoNoise, **default**=0

Property MotionDetector\_ReduceVideoNoise As Boolean

### Description

When enabled, reduces the sensitivity to the video noise.  
Enabling this property limits the risk of motion detected because of the video noise.  
This processing increases the CPU load.

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_Reset](#)



[MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Produce online help for Qt applications](#)

## MotionDetector\_Triggered

### TVideoGrabber.MotionDetector\_Triggered

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the triggered motion detection.

#### Declaration

**property** MotionDetector\_Triggered: Boolean **read** GetMotionDetector\_Triggered **write** SetMotionDetector\_Triggered **default** DEF\_MotionDetector\_Triggered;

\_\_property **bool** MotionDetector\_Triggered=read=GetMotionDetector\_Triggered, write=SetMotionDetector\_Triggered, **default**=0

Property MotionDetector\_Triggered As Boolean

#### Description

Used to enable the triggered motion detection.

By default the motion detection occurs for each video frame.

Unlikely, when this property is enabled, the motion detection will occur only one time when [MotionDetector\\_TriggerNow](#) is invoked, until the next [MotionDetector\\_TriggerNow](#) call.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Easily create CHM Help documents](#)

## MouseWheelEventEnabled

### TVideoGrabber.MouseWheelEventEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Activates the OnMouseWheel event

#### Declaration

**property** MouseWheelEventEnabled: Boolean **read** GetMouseWheelEventEnabled **write** SetMouseWheelEventEnabled **default** DEF\_MouseWheelEventEnabled;



**\_\_property bool** MouseWheelEventEnabled==GetMouseWheelEventEnabled,  
write=SetMouseWheelEventEnabled

Property MouseWheelEventEnabled as Boolean

### Description

Enable it to activate the OnMouseWheel event when the user turns the mouse wheel up or down over the video window.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## MpegStreamType

### TVideoGrabber.MpegStreamType

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Type of the stream.

### Declaration

**property** MpegStreamType: TMpegStreamType **read** GetMpegStreamType **write** SetMpegStreamType;

**\_\_property** TMpegStreamType MpegStreamType=read=GetMpegStreamType, write=SetMpegStreamType,  
**nodefault**

Property MpegStreamType As TMpegStreamType

### Description

[TMpegStreamType](#) type of the stream when the video capture device outputs an MPEG stream (when IsMPEGStream returns true).

This property is meaningful only when the video capture device is a MPEG device IsMPEGStream IsMPEGDevice returns true.

Type of the Mpeg stream. The current [video capture device](#) must be [previewing](#) or [capturing](#) .

Created with the Standard Edition of HelpNDoc: [Full-featured Documentation generator](#)

## MultiplexedInputEmulation

### TVideoGrabber.MultiplexedInputEmulation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Automatically switches inputs periodically.

#### Declaration

**property** MultiplexedInputEmulation: Boolean **read** GetMultiplexedInputEmulation **write** SetMultiplexedInputEmulation;

\_\_property **bool** MultiplexedInputEmulation =read=GetMultiplexedInputEmulation, write=SetMultiplexedInputEmulation, **nodefault**

Property MultiplexedInputEmulation As Boolean

#### Description

Used to switch inputs periodically in multiplexed mode.

If the video capture board does not switch the inputs automatically, enable this property to get the inputs switched programmatically in the following frame order:

1...2...3...4...1...2...3...4...1...2...3...4... and so on.

See the [Video capture devices having multiplexed inputs](#) chapter for more information.

#### See Also

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedRole](#) [MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [Easy EBook and documentation generator](#)

## MultiplexedRole

### TVideoGrabber.MultiplexedRole

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the component role with multiplexed video inputs.

#### Declaration

**property** MultiplexedRole: TMultiplexedRole **read** GetMultiplexedRole **write** SetMultiplexedRole **default** DEF\_MultiplexedRole;

\_\_property TMultiplexedRole MultiplexedRole=read=GetMultiplexedRole, write=SetMultiplexedRole, **default=0**

Property MultiplexedRole As TxMultiplexedRole

#### Description

This [TMultiplexedRole](#) property is used to specify the role of the component with BT878-based video cards having multiplexed inputs.

mr\_NotMultiplexed : normal mode

mr\_MultiplexedMosaic4 : displays the 4 inputs in a "mosaic layout", the video size is 2 times the normal size

mr\_MultiplexedMosaic16 : displays the 16 inputs in a "mosaic layout", the video size is 4 times the normal size

mr\_MultiplexedMaster : sets the TVideoGrabber as master (it will distribute the inputs to the slaves)  
 mr\_MultiplexedSlave : sets the TVideoGrabber component as slave (it will receive a virtual input from the master).

See the [multiplexed video inputs](#) chapter.

#### See Also

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#)  
[MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## MultiplexedStabilizationDelay

### TVideoGrabber.MultiplexedStabilizationDelay

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the delay to wait for the image stabilization.

#### Declaration

**property** MultiplexedStabilizationDelay: LongInt **read** GetMultiplexedStabilizationDelay **write** SetMultiplexedStabilizationDelay **default** DEF\_MultiplexedStabilizationDelay;

\_\_property **int** MultiplexedStabilizationDelay = **read**=GetMultiplexedStabilizationDelay, **write**=SetMultiplexedStabilizationDelay

Property MultiplexedStabilizationDelay As Long

#### Description

Used to set/retrieve the delay to wait for the image stabilization, expressed in milliseconds.

Depending of the video capture board, a various time is required for the input to be properly switched and the image stabilized.

If you get unstable video display cross-channel mixing, try to increase this value (default value = 70 milliseconds)

I too low value may cause jumping frames or cross-input mixed fames.

#### See Also

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#)  
[MultiplexedRole](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## MultiplexedSwitchDelay

### TVideoGrabber.MultiplexedSwitchDelay

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the time each input is active, before switching to the next input.

#### Declaration

**property** MultiplexedSwitchDelay: LongInt **read** GetMultiplexedSwitchDelay **write** SetMultiplexedSwitchDelay **default** DEF\_MultiplexedSwitchDelay;

\_\_property **int** MultiplexedSwitchDelay = **read**=GetMultiplexedSwitchDelay, **write**=SetMultiplexedSwitchDelay

Property MultiplexedSwitchDelay As Long

### Description

Used to specify/retrieve the time each input is active, before switching to the next input, expressed in number of frames.

Default value: 0 (means the input is switched after each video frame).

### See Also

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#) [MultiplexedRole](#) [MultiplexedStabilizationDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## MuteAudioRendering

### TVideoGrabber.MuteAudioRendering

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Mutes the audio speakers.

### Declaration

**property** MuteAudioRendering: Boolean **read** GetMuteAudioRendering **write** SetMuteAudioRendering **default** DEF\_MuteAudioRendering;

\_\_property bool MuteAudioRendering=read=GetMuteAudioRendering, write=SetMuteAudioRendering, **default**=0

Property MuteAudioRendering As Boolean

### Description

Used to mute the audio speakers.

### See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

## NDIFormatType

### TVideoGrabber.NDIFormatType

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Type of NDI streaming

### Declaration

**property** NDIFormatType: TNDIFormatType **read** GetNDIFormatType **write** SetNDIFormatType **default** DEF\_NDIFormatType;

\_\_property TNDIFormatType NDIFormatType==GetNDIFormatType, write=SetNDIFormatType, **default**=1;

### Description

Type of the NDI streaming format

### See Also

[TNDIFormatType](#) [NDIName](#) [NetworkStreaming](#)

## NDIName

### TVideoGrabber.NDIName

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Name of the NDI streaming

#### Declaration

**property** NDIName: **string** **read** GetNDIName **write** SetNDIName;

\_\_**property** System::UnicodeString NDIName==GetNDIName, write=SetNDIName;

#### Description

Name of the NDI streaming

See [Streaming using Newtek NDI](#)

#### See Also

[TNDIFormatType](#) [NDIFormatType](#) [NetworkStreaming](#)

## NetworkStreaming

### TVideoGrabber.NetworkStreaming

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Enables the network streaming.

#### Declaration

**property** NetworkStreaming: TNetworkStreaming **read** GetNetworkStreaming **write** SetNetworkStreaming  
**default** DEF\_NetworkStreaming;

\_\_**property** TNetworkStreaming NetworkStreaming=read=GetNetworkStreaming,  
write=SetNetworkStreaming, **default**=0

Property NetworkStreaming As TxNetworkStreaming

#### Description

This [TNetworkStreaming](#) property is used to enable the live network streaming when the [recording](#) is running.

Allowed values:

**ns\_Disabled** network streaming is disabled

**ns\_ASFDirectNetworkStreaming** direct network streaming from the platform running TVideoGrabber

**ns\_ASFStreamingToPublishingPoint** TVideoGrabber sends the streaming media to the Windows Media Server specified by [ASFMediaServerPublishingPoint](#).

**ns\_NDI** Streaming to NDI (requires the NDI Runtime 6 or newer to be installed)

#### See Also

[TNDIFormatType](#) [TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#)  
[TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#)  
[ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#)  
[ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#)  
[ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#)

[ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NDIFormatType](#) [NDIName](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Streamline Your Documentation Process with HelpNDoc's Intuitive Interface

## NetworkStreamingType

### TVideoGrabber.NetworkStreamingType

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the type of streaming (audio, video or both)

#### Declaration

**property** NetworkStreamingType: TNetworkStreamingType **read** GetNetworkStreamingType **write** SetNetworkStreamingType **default** DEF\_NetworkStreamingType;

\_\_property TNetworkStreamingType NetworkStreamingType=read=GetNetworkStreamingType, write=SetNetworkStreamingType, **default**=0

Property NetworkStreamingType As TxNetworkStreamingType

#### Description

Used to specify the type of streaming when [NetworkStreaming](#) <> ns\_Disabled.

#### Values:

nst\_AudioVideoStreaming : audio + video streaming  
nst\_VideoStreaming : video streaming only  
nst\_AudioStreaming : audio streaming only

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: Easily create CHM Help documents

## NormalCursor

### TVideoGrabber.NormalCursor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Cursor displayed when a TVideoGrabber's task ends.

#### Declaration

**property** NormalCursor: TCursor **read** GetNormalCursor **write** SetNormalCursor **default** DEF\_NormalCursor;

\_\_property Controls::TCursor NormalCursor=read=GetNormalCursor, write=SetNormalCursor, **default**=0

Property NormalCursor As TxCursors

## Description

Cursor displayed when a TVideoGrabber's task ends when [BusyCursor](#) <> crDefault.  
If [BusyCursor](#) = crDefault, NormalCursor is ignored.

Created with the Standard Edition of HelpNDoc: [Make Help Documentation a Breeze with a Help Authoring Tool](#)

## NotificationMethod

### TVideoGrabber.NotificationMethod

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies if the events are notified by timer or by thread.

#### Declaration

**property** NotificationMethod: TNotificationMethod **read** GetNotificationMethod **write** SetNotificationMethod  
**default** DEF\_NotificationMethod;

\_\_property TNotificationMethod NotificationMethod=read=GetNotificationMethod,  
write=SetNotificationMethod, **default**=0

Property NotificationMethod As TxNotificationMethod

#### Description

#### DEPRECATED

*This [TNotificationMethod](#) property is used to specify if the events must be notified by timer or by thread.*

*The default method is by timer (nm\_Timer), however depending of your application environment, you can prefer using a notification by thread (nm\_Thread).*

*You can switch the method on the fly, event during preview, recording or playback.*

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## NotificationPriority

### TVideoGrabber.NotificationPriority

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the thread priority when the events are notified by thread.

#### Declaration

**property** NotificationPriority: TThreadPriority **read** GetNotificationPriority **write** SetNotificationPriority  
**default** DEF\_NotificationPriority;

\_\_property Classes::TThreadPriority NotificationPriority=read=GetNotificationPriority,  
write=SetNotificationPriority, **default**=3

Property NotificationPriority As TxThreadPriority

#### Description

Used to Specify the thread priority when the events are notified by thread (when [NotificationMethod](#) = nm\_Thread).



## NotificationSleepTime

### TVideoGrabber.NotificationSleepTime

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the sleep time between notifications.

#### Declaration

**property** NotificationSleepTime: LongInt **read** GetNotificationSleepTime **write** SetNotificationSleepTime  
**default** DEF\_NotificationSleepTime;

\_\_property int NotificationSleepTime=read=GetNotificationSleepTime, write=SetNotificationSleepTime,  
**default**=- 1

Property NotificationSleepTime As Long

#### Description

Used to specify the sleep time between event notifications (in milliseconds).

-1 = default value.

E.g. you can assign e.g. 200 to limit the notifications to 5 per second. In this case the [OnFrameProgress](#) event will occurs only 5 times per second, even if the frame rate is 30 fps.

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## OnFrameBitmapEventSynchron

### TVideoGrabber.OnFrameBitmapEventSynchron

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Make the OnFrameBitmap event callback synchron

#### Declaration

**property** OnFrameBitmapEventSynchron: Boolean **read** GetOnFrameBitmapEventSynchron **write** SetOnFrameBitmapEventSynchron **default** DEF\_OnFrameBitmapEventSynchron;

\_\_property bool OnFrameBitmapEventSynchron=read=GetOnFrameBitmapEventSynchron,  
write=SetOnFrameBitmapEventSynchron, **default**=0

Property OnFrameBitmapEventSynchron As Boolean

#### Description

When enabled, synchronize the OnFrameBitmap event callback with the main thread, allowing to update Windows controls or perform blocking operations from the event.

**Warning: enabling this event can slow down the application or introduce latency in the video stream.**

#### See Also

[TCardinalDirection](#) [TFrameGrabberRGBFormat](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [FrameGrabber](#)  
[FrameGrabberCurrentRGBFormat](#) [FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#)  
[GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [InFrameProgressEvent](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#)  
[OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#)  
[SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)

[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TranslateMouseCoordinates](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: Revolutionize your documentation process with HelpNDoc's online capabilities

## OpenURLAsync

### TVideoGrabber.OpenURLAsync

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

enables / disables the asynchronous URL connection

#### Declaration

**property** OpenURLAsync: Boolean **read** GetOpenURLAsync **write** SetOpenURLAsync **default** DEF\_OpenURLAsync;

\_\_**property bool** OpenURLAsync=read=GetOpenURLAsync, write=SetOpenURLAsync, **default=0**

Property OpenURLAsync As Boolean

#### Description

Specifies if the component must connect to IP cameras or URLs asynchronously or synchronously.

See [Asynchronous vs Asynchronous connection](#)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#)  
[GetLastErrorMessages](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#)  
[OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#)  
[ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#)  
[OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: Free help authoring tool

## OverlayAfterTransform

### TVideoGrabber.OverlayAfterTransform

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies if the overlays are performed before or after the transforms

**Declaration**

**property** OverlayAfterTransform: Boolean **read** GetOverlayAfterTransform **write** SetOverlayAfterTransform  
**default** DEF\_OverlayAfterTransform;

**\_\_property** bool OverlayAfterTransform=read=GetOverlayAfterTransform, write=SetOverlayAfterTransform,  
**default**=0

Property OverlayAfterTransform As Boolean

**Description**

Used to specify if the graphics and text overlays are performed before or after the transforms (like video rotation, cropping, etc...).

Default value : false (overlays before transforms).

When this property is enabled and [VideoProcessing Rotation](#) is used, any drawing performed from the [OnFrameOverlayUsingDC](#) event will be rotated.

To retrieve the initial coordinates and prevent the drawing to be rotated, use the [RetrieveInitialXYAfterRotation](#).

**See Also**

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#)  
[SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#)  
[SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#)  
[SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#)  
[SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#)  
[SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#)  
[SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#)  
[SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

**ParentWindow****TVideoGrabber.ParentWindow**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the window handle that underlies the parent control.

**Declaration**

**property** ParentWindow: HWND **read** GetParentWindow **write** SetParentWindow;

\_\_property HWND ParentWindow=read=GetParentWindow, write=SetParentWindow, **ndefault**

n/a

### Description

ParentWindow refers to the window handle that underlies the parent control. To designate a non-VCL control as a parent, assign that control's handle to ParentWindow.

Setting ParentWindow has no effect if the **Parent** property is not nil.

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## PlayerAudioCodec

### TVideoGrabber.PlayerAudioCodec

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Audio codec currently used by the player.

### Declaration

**property** PlayerAudioCodec: **String** read GetPlayerAudioCodec;

\_\_property wchar\_t \*PlayerAudioCodec=read=GetPlayerAudioCodec

Property PlayerAudioCodec As String

### Description

Returns the name of the audio codec currently used by the player, if any.  
The clip must have been opened by [OpenPlayer](#).

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

## PlayerAudioRendering

### TVideoGrabber.PlayerAudioRendering

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Disables the audio rendering during playback, and allows to use wider speed ratio ranges.

### Declaration

**property** PlayerAudioRendering: Boolean **read** GetPlayerAudioRendering **write** SetPlayerAudioRendering **default** DEF\_PlayerAudioRendering;

\_\_property **bool** PlayerAudioRendering=read=GetPlayerAudioRendering, write=SetPlayerAudioRendering, default=1

Property PlayerAudioRendering As Boolean

### Description

This property allows to enable/disable the audio rendering during playback of video clips.

When audio is disabled, it is possible to use wider speed ratio ranges. See [PlayerSpeedRatio](#) .

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Protect Your Confidential PDFs with These Simple Security Measures](#)

## PlayerDuration

### TVideoGrabber.PlayerDuration

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the total duration of an opened video clip.

### Declaration

**property** PlayerDuration: LargeInteger **read** GetPlayerDuration **write** SetPlayerDuration;

\_\_property \_\_int64 PlayerDuration=read=GetPlayerDuration write=SetPlayerDuration

Property PlayerDuration As Double

### Description

Used to retrieve the total duration of a video clip opened by [OpenPlayer](#) (expressed in in 100-nanosecond units, divide it by 10000000 to convert in seconds).

This property is theoretically a read-only property, however you can assign a value to this property to specify the display time when opening static images in the player (useful mainly when opening static images in the [Playlist](#)).

E.g. if the playlist must display each static image during 4 seconds, set PlayerDuration = 40000000 before invoking [Playlist](#) (pl\_Play, "").

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#)

[PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## PlayerDVSize

### TVideoGrabber.PlayerDVSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the video size when playing back DV-encoded video clips.

#### Declaration

**property** PlayerDVSize: TDVSize **read** GetPlayerDVSize **write** SetPlayerDVSize **default** DEF\_PlayerDVSize;

\_\_property TDVSize PlayerDVSize=read=GetPlayerDVSize, write=SetPlayerDVSize, **default**=0

Property PlayerDVSize As TxDVSize

#### Description

Used to specify the video size when playing back DV-encoded video clips.  
The value is a [TDVSize](#) type.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## PlayerFastSeekSpeedRatio

### TVideoGrabber.PlayerFastSeekSpeedRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Speed ratio when playing a video clip at different speeds, forwards or backwards.

#### Declaration

**property** PlayerFastSeekSpeedRatio: LongInt **read** GetPlayerFastSeekSpeedRatio **write** SetPlayerFastSeekSpeedRatio **default** DEF\_PlayerFastSeekSpeedRatio;

\_\_property **int** PlayerFastSeekSpeedRatio=read=GetPlayerFastSeekSpeedRatio, write=SetPlayerFastSeekSpeedRatio, **default**=4



Property PlayerFastSeekSpeedRatio As Long

### Description

Set or retrieves the speed ratio used to play a video clip at different speeds forwards or backwards using [RewindPlayer](#) or [FastForwardPlayer](#) .

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## PlayerFileName

### TVideoGrabber.PlayerFileName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the full qualified path and file name of a video clip.

### Declaration

**property** PlayerFileName: **string** **read** GetPlayerFileName **write** SetPlayerFileName;

\_\_property wchar\_t \*PlayerFileName=read=GetPlayerFileName, write=SetPlayerFileName

Property PlayerFileName As String

### Description

Used to set or retrieve the full qualified path and file name of a video clip that will be opened then by [OpenPlayer](#) .

E.g.:

```
VideoGrabber.PlayerFileName = "myvideoclip.avi"
VideoGrabber.AutoStartPlayer = false
VideoGrabber.OpenPlayer()
```

### Remarks:

- to open a DVD at a specified location, set location of the the VIDEO\_TS folder, e.g.:

```
VideoGrabber.PlayerFileName = "E:
VideoGrabber.OpenPlayer()
```

- to open the default DVD, set an empty string and invoke OpenDVD:

```
VideoGrabber.PlayerFileName = ""
VideoGrabber.OpenDVD()
```

### See Also



[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [DVDInfo](#) [DVDTitle](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenDVD](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## PlayerForcedCodec

### TVideoGrabber.PlayerForcedCodec

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Force the use of a given codec during playback.

#### Declaration

**property** PlayerForcedCodec: **string** **read** GetPlayerForcedCodec **write** SetPlayerForcedCodec;

\_\_property wchar\_t \*PlayerForcedCodec=read=GetPlayerForcedCodec, write=SetPlayerForcedCodec

Property PlayerForcedCodec As String

#### Description

Used to specify particular codecs to be used by TVideoGrabber.

TVideoGrabber automatically selects the default appropriate codec to play a given clip.

However in some cases a different codec can be preferred. E.g. to play MPEG2 clips you can prefer the Elecard/Moonlight video decoder instead of the Intervideo video decoder.

*Note that adding codecs to the graph can be achieved also by invoking [ThirdPartyFilter\\_AddToList](#) (tpf\_AddToGraph...*

To use the default codecs, leave **PlayerForcedCodec** empty.

To force the use of a given codec, simply assign to this property:

- either a significant part of the codec name,
- either the CLSID string of the codec.

E.g.:

`VideoGrabber1.PlayerForcedCodec = "Elecard MPEG2 Video Decoder"`

or

`VideoGrabber1.PlayerForcedCodec = "F50B3F13-19C4-11CF-AA9A-02608C9BABA2"`

#### PlayerForcedCodec keywords

The following keywords are supported by the PlayerForcedCodec property:

PlayerForcedCodec = "**NOFFDSHOW**"

Prevents TVideoGrabber to use the FFDSHOW codecs when they are enabled by default on the current platform.

PlayerForcedCodec = "**NOMPC**"

Prevents TVideoGrabber to use the MPC (Media Player Classic) codecs when they are enabled by default on the current platform.

PlayerForcedCodec = "AVS"

enables the playback through AVISynth when [AVISynth](#) is installed

PlayerForcedCodec = "FFDSHOW"

enables the playback through FFDSHOW is the FFDSHOW video decoder is configured to decode the format to be played.

All these non-default settings can be restored to their default value by setting PlayerForcedCodec = ""

*Note that reading the PlayerForcedCodec string does not reflect the state of these keywords, only codec names or codec CLSIDs appear in the PlayerForcedCodec string.*

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

## PlayerFrameCount

### TVideoGrabber.PlayerFrameCount

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the total number of video frames of an opened video clip.

#### Declaration

**property** PlayerFrameCount: LargeInteger **read** GetPlayerFrameCount;

\_\_property \_\_int64 PlayerFrameCount=read=GetPlayerFrameCount

Property PlayerFrameCount As Double

#### Description

Used to retrieve the total number of video frames of a clip opened by [OpenPlayer](#) .

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

## PlayerFramePosition

### TVideoGrabber.PlayerFramePosition

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the current frame position of an opened video clip.

#### Declaration

**property** PlayerFramePosition: LargeInteger **read** GetPlayerFramePosition **write** SetPlayerFramePosition;

\_\_property \_\_int64 PlayerFramePosition=read=GetPlayerFramePosition, write=SetPlayerFramePosition

Property PlayerFramePosition As Double

#### Description

Used to retrieve the current frame position of a clip opened by [OpenPlayer](#) .

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

## PlayerFrameRate

### TVideoGrabber.PlayerFrameRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the frame rate of a video clip.

#### Declaration

**property** PlayerFrameRate: Double **read** GetPlayerFrameRate;

\_\_property **double** PlayerFrameRate=read=GetPlayerFrameRate

Property PlayerFrameRate As Double

#### Description

Used to retrieves the frame rate (expressed in frames per second) of a video clip opened with [OpenPlayer](#) .

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#)

[PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#)  
[PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#)  
[PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#)  
[PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#)  
[RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#)  
[SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

## PlayerHwAccel

### TVideoGrabber.PlayerHwAccel

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Hardware video acceleration

#### Declaration

**property** PlayerHwAccel: THwAccel **read** GetPlayerHwAccel **write** SetPlayerHwAccel **default** DEF\_PlayerHwAccel;

**\_\_property** THwAccel PlayerHwAccel=read=GetPlayerHwAccel, write=SetPlayerHwAccel, **default**=hw\_None

Property PlayerHwAccel As THwAccel

#### Description

Enables the hardware acceleration of video decoding during the playback  
 This feature is supported through the [LAV filters](#) that must be installed first.  
 The possible values are enumerated here: [THwAccel](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## PlayerOpenProgressPercent

### TVideoGrabber.PlayerOpenProgressPercent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Percentage of the video being opened in the background

#### Declaration

**property** PlayerOpenProgressPercent: LongInt **read** GetPlayerOpenProgressPercent: LongInt;

**\_\_property** int PlayerOpenProgressPercent==GetPlayerOpenProgressPercent, **nodefault**};

property PlayerOpenProgressPercent as Long

#### Description

Used to retrieve the percentage of the video being opened in the background

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## PlayerRefreshPausedDisplay

### TVideoGrabber.PlayerRefreshPausedDisplay

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

**ber**

Specifies if the video window must be refreshed when the video clip is paused.

**Declaration**

**property** PlayerRefreshPausedDisplay: THwAccel **read** GetPlayerRefreshPausedDisplay **write** SetPlayerRefreshPausedDisplay **default** DEF\_PlayerRefreshPausedDisplay;

\_\_**property bool** PlayerRefreshPausedDisplay=read=GetPlayerRefreshPausedDisplay, write=SetPlayerRefreshPausedDisplay, **default**=0

Property PlayerRefreshPausedDisplay As Boolean

**Description**

If enabled, when the video clip is pause the video window is refreshed at a frame rate specified by [PlayerRefreshPausedDisplayFrameRate](#) property.

Enable this feature to get the graphics and text overlays refreshed when the video clip is paused.

**See Also**

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

**PlayerRefreshPausedDisplayFrameRate****TVideoGrabber.PlayerRefreshPausedDisplayFrameRate**
[Prev](#)
[Next](#)
**TVideoGrabber** **Properties**

Specifies the frame rate used to refresh the display when a video clip is paused.

**Declaration**

**property** PlayerRefreshPausedDisplayFrameRate: Double **read** GetPlayerRefreshPausedDisplayFrameRate **write** SetPlayerRefreshPausedDisplayFrameRate;

\_\_**property double**

PlayerRefreshPausedDisplayFrameRate=read=GetPlayerRefreshPausedDisplayFrameRate, write=SetPlayerRefreshPausedDisplayFrameRate

Property PlayerRefreshPausedDisplayFrameRate As Double

**Description**

Specifies the frame rate used to refresh periodically the display when a video clip is in a paused state and the [PlayerRefreshPausedDisplay](#) property is enabled.

If PlayerRefreshPausedDisplayFrameRate = 0, the display is refreshed at about 30 fps.

E.g.:

PlayerRefreshPausedDisplayFrameRate = 1 refreshes the paused display every second

PlayerRefreshPausedDisplayFrameRate = 5 refreshes the paused display 5 times per second

PlayerRefreshPausedDisplayFrameRate = 0.25 refreshes the paused display every 4 seconds.

## See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## PlayerSpeedRatio

### TVideoGrabber.PlayerSpeedRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the playback speed ratio.

#### Declaration

**property** PlayerSpeedRatio: Double **read** GetPlayerSpeedRatio **write** SetPlayerSpeedRatio;

\_\_property **double** PlayerSpeedRatio=read=GetPlayerSpeedRatio, write=SetPlayerSpeedRatio

Property PlayerSpeedRatio As Double

#### Description

Used to set or retrieve the player's speed ratio when playing a clip by using [OpenPlayer](#) and/or [RunPlayer](#) .

The value is expressed as a ratio of the normal speed, e.g. 0.5 is half seed and 2 is twice speed.

- if the player audio rendering is enabled ([PlayerAudioRendering](#) = true), the speed ratio can be usually used in the 0.5 to 2 range.
- if the player audio rendering is disabled ([PlayerAudioRendering](#) = false), a larger speed ratio range can be used (from 0.1 to 5 or more).

## See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## PlayerSpeedRatioConstantAudioPitch

### TVideoGrabber.PlayerSpeedRatioConstantAudioPitch

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)



**ber**

Keeps a constant audio pitch for speed ratio  $\neq 1$

**Declaration**

**property** PlayerSpeedRatioConstantAudioPitch: Double **read** GetPlayerSpeedRatioConstantAudioPitch  
**write** SetPlayerSpeedRatioConstantAudioPitch;

\_\_**property double** PlayerSpeedRatioConstantAudioPitch=read=GetPlayerSpeedRatioConstantAudioPitch,  
write=SetPlayerSpeedRatioConstantAudioPitch

Property PlayerSpeedRatioConstantAudioPitch As Double

**Description**

When [PlayerSpeedRatio](#)  $\neq 1$ :

- if PlayerSpeedRatioConstantAudioPitch is enabled (default), a constant audio pitch is maintained (audio samples are dropped)
- if PlayerSpeedRatioConstantAudioPitch is disabled, the audio pitch is increased or decreased according to the player speed.

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

**PlayerState****TVideoGrabber.PlayerState**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)  
**ber**

Returns the current player state.

**Declaration**

**property** PlayerState: TPlayerState **read** GetPlayerState;

\_\_**property** TPlayerState PlayerState==GetPlayerState, **nodefault**};

property PlayerState as TPlayerState

**Description**

Used to retrieve the current player [state](#).

The possible values are:

ps\_Closed  
ps\_Stopped  
ps\_Paused  
ps\_Playing  
ps\_PlayingBackward  
ps\_FastForwarding  
ps\_FastRewinding

Each value is greater than the previous one, so it is possible to test e.g. if the clip is playing in any direction by testing PlayerState  $\geq$  ps\_Playing.

Created with the Standard Edition of HelpNDoc: [Effortlessly create a professional-quality documentation website with HelpNDoc](#)

**PlayerTimePosition****TVideoGrabber.PlayerTimePosition**[Prev](#)[Next](#)



## **TVideoGrabber** Properties

Sets or retrieves the current time position of a video clip.

### **Declaration**

**property** PlayerTimePosition: LargeInteger **read** GetPlayerTimePosition **write** SetPlayerTimePosition;

\_\_property \_\_int64 PlayerTimePosition=read=GetPlayerTimePosition, write=SetPlayerTimePosition

Property PlayerTimePosition As Double

### **Description**

Used to set or retrieve the current time position of a video clip opened with [OpenPlayer](#) (expressed in in 100-nanosecond units, divide it by 10000000 to convert in seconds).

### **See Also**

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## **PlayerTrackBar**

### **TVideoGrabber.PlayerTrackBar**

[Prev](#)
[Next](#)

## **TVideoGrabber** Properties

Associates a trackbar to the player (Delphi and C++Builder versions only).

### **Declaration**

**property** PlayerTrackBar: TTrackBar **read** GetPlayerTrackBar **write** SetPlayerTrackBar **default** DEF\_PlayerTrackBar;

\_\_property ComCtrls::TTrackBar \*PlayerTrackBar=read=GetPlayerTrackBar, write=SetPlayerTrackBar, default=0

n/a

### **Description**

Used to associate a trackbar to the player (Delphi and C++Builder versions only).

To use it:

- put a trackbar component on the form,
- select the TVideoGrabber component with the object inspector,
- browse the PlayerTrackBar property and select the trackbar you put on the form.

At runtime, when opening a video clip, the trackbar frequency is set to the number of frames of the video clip (1).

During playback, the trackbar cursor moves automatically, according to the current position of the video clip played.  
When the clip is paused, you can set a new position simply by moving the trackbar cursor.

(1) note: the frequency of the trackbar is limited to 32768, see [PlayerTrackBarScale](#) for more information.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Free Kindle producer

## PlayerTrackBarScale

### TVideoGrabber.PlayerTrackBarScale

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Scale of the trackbar cursor frequency (Delphi and C++Builder versions only).

#### Declaration

**property** PlayerTrackBarScale: Double **read** GetPlayerTrackBarScale;

\_\_\_property **double** PlayerTrackBarScale =read=GetPlayerTrackBarScale

n/a

#### Description

Scale of the trackbar cursor frequency, when the number of frames of the video clip is greater than 32768 (Delphi and C++Builder versions only).

When opening a video clip:

- if the number of frames of the video clip is lower or equal to 32768, the frequency of the trackbar is set to the number of frames of the video clip, and PlayerTrackBarScale = 1
- if the number of frames of the video clip is greater than 32768, the frequency of the trackbar is set to 32768, and PlayerTrackBarScale is set to the number of frames / 32768.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

## PlayerTrackBarSynchronone

### TVideoGrabber.PlayerTrackBarSynchronone

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Enable the synchronone behavior of the trackbar

#### Declaration

**property** PlayerTrackBarSynchronone: Boolean **read** GetPlayerTrackBarSynchronone **write** SetPlayerTrackBarSynchronone **default** DEF\_PlayerTrackBarSynchronone;

**\_\_property** bool PlayerTrackBarSynchronone=read=GetPlayerTrackBarSynchronone, write=SetPlayerTrackBarSynchronone, **default**=0

Property PlayerTrackBarSynchronone as Boolean

#### Description

When this property is enabled, moving the trackbar with the mouse moves also the position in the video clip immediately.

When this property is disabled, the position is updated only after moving the mouse on the trackbar, when the mouse button is released.

#### Important:

For this feature to work property you must inform TVideoGrabber of the mouse and keyboard actions on your trackbar by invoking [NotifyPlayerTrackbarAction](#) from your trackbar's event.

*Look at the trackbar's code of the MainDemo project for sample code.*

When using Delphi or C++Builder, if a TTrackbar component is assigned to the [PlayerTrackbar](#) property this is handled automatically.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

## PlayerVideoCodec

### TVideoGrabber.PlayerVideoCodec

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Video codec currently used by the player.

#### Declaration

**property** PlayerVideoCodec: **String** **read** GetPlayerVideoCodec;

\_\_property wchar\_t \*PlayerVideoCodec=read=GetPlayerVideoCodec

Property PlayerVideoCodec As String

### Description

Returns the name of the video codec currently used by the player, if any.  
The clip must have been opened by [OpenPlayer](#).

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## PlaylistIndex

### TVideoGrabber.PlaylistIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the current position in the playlist.

### Declaration

function GetPlaylistIndex: LongInt;

int GetPlaylistIndex(void);

Function GetPlaylist as Long

### Description

Returns the position of the clip currently played in the playlist.

See the "[Using the playlist](#)" chapter for more information about the playlist feature.

### See Also

[TPlaylist](#) [Video formats](#) [GetPlaylist](#) [IsPlaylistActive](#) [OnPlayerEndOfPlaylist](#) [Playlist](#)

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

## PreallocCapFileCopiedAfterRecording

### TVideoGrabber.PreallocCapFileCopiedAfterRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables recopy of the preallocated file immediately after capture.

### Declaration

**property** PreallocCapFileCopiedAfterRecording: Boolean **read** GetPreallocCapFileCopiedAfterRecording

**write** SetPreallocCapFileCopiedAfterRecording **default** DEF\_PreallocCapFileCopiedAfterRecording;

\_\_property **bool** PreallocCapFileCopiedAfterRecording=read=GetPreallocCapFileCopiedAfterRecording, write=SetPreallocCapFileCopiedAfterRecording, **default**=1

Property PreallocCapFileCopiedAfterRecording As Boolean

### Description

This property is significant only when the preallocated file is used ("[PreallocCapFileEnabled](#)" is true).

- when this property is **enabled**, the recopy of the preallocated file occurs immediately after [StopRecording](#) is invoked. ***This is the normal default value.***

- when this property is **disabled**, the preallocated file is not recopied.

This option is useful only if you prefer to keep the video clip "as is" in the preallocated capture file and if you plan to recopy or reencode it later by using [StartReencoding](#).

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface

## PreallocCapFileEnabled

### TVideoGrabber.PreallocCapFileEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to pre-allocate the recording file.

### Declaration

**property** PreallocCapFileEnabled: Boolean **read** GetPreallocCapFileEnabled **write** SetPreallocCapFileEnabled **default** DEF\_PreallocCapFileEnabled;

\_\_property **bool** PreallocCapFileEnabled=read=GetPreallocCapFileEnabled, write=SetPreallocCapFileEnabled, **default**=0

Property PreallocCapFileEnabled As Boolean

### Description

If enabled, a preallocated capture file is created before starting recording.

See the "**preallocated capture file**" section in the [AVI recording](#) chapter for more information.

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#)

[AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

## PreallocCapFileName

### TVideoGrabber.PreallocCapFileName

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the full qualified path and name of the pre-allocated recording file.

#### Declaration

**property** PreallocCapFileName: **string** **read** GetPreallocCapFileName **write** SetPreallocCapFileName;

\_\_property wchar\_t \*PreallocCapFileName=read=GetPreallocCapFileName, write=SetPreallocCapFileName

Property PreallocCapFileName As String

#### Description

Used to set or retrieve the full qualified path and name of the pre-allocated recording file.

This property is ignored if [PreallocCapFileEnabled](#) is disabled.

If left blank, a "prealloc.avi" file will be created in the [StoragePath](#) directory.

See the "**preallocated capture file**" section in the [AVI recording](#) chapter for more information.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

## PreallocCapFileSizeInMB

### TVideoGrabber.PreallocCapFileSizeInMB

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)



**ber**

Size of the pre-allocated recording file, in Mb.

**Declaration**

**property** PreallocCapFileSizeInMB: LongInt **read** GetPreallocCapFileSizeInMB **write** SetPreallocCapFileSizeInMB **default** DEF\_PreallocCapFileSizeInMB;

\_\_property int PreallocCapFileSizeInMB=read=GetPreallocCapFileSizeInMB, write=SetPreallocCapFileSizeInMB, **default**=100

Property PreallocCapFileSizeInMB As Long

**Description**

Used to set or retrieve the size of the pre-allocated recording file, in Mb.

For best capture results, capture to a unfragmented, pre-allocated capture file that is larger than the size of the capture data.

This property is ignored if [PreallocCapFileEnabled](#) is disabled.

See the "**preallocated capture file**" section in the [AVI recording](#) chapter for more information.

**See Also**

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Easily share your documentation with the world through a beautiful website

**PreviewZoomSize****TVideoGrabber.PreviewZoomSize**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Modifies the size of the preview window (in percent of the [native video width](#) and [height](#) ).

**Declaration**

**property** PreviewZoomSize: LongInt **read** GetPreviewZoomSize **write** SetPreviewZoomSize **default** DEF\_PreviewZoomSize;

\_\_property int PreviewZoomSize=read=GetPreviewZoomSize, write=SetPreviewZoomSize, **default**=100

Property PreviewZoomSize As Long

**Description**

Used to modify the size of the preview window (in percent of the [native video width](#) and [height](#) ).  
Default value: 100 (percent).

**See Also**



[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

## RawAudioSampleCapture

### TVideoGrabber.RawAudioSampleCapture

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the capture of the raw audio samples.

#### Declaration

**property** RawAudioSampleCapture: Boolean **read** GetRawAudioSampleCapture **write** SetRawAudioSampleCapture **default** DEF\_RawAudioSampleCapture;

\_\_property **bool** RawAudioSampleCapture=read=GetRawAudioSampleCapture, write=SetRawAudioSampleCapture, **default**=0

Property RawAudioSampleCapture As Boolean

#### Description

Used to enable / disable the capture of each raw audio sample coming out of the audio source. Each audio sample will be returned by the [OnRawAudioSample](#) event.

#### See Also

[TOnRawSample](#) [OnRawAudioSample](#) [OnRawVideoSample](#) [RawCaptureAsyncEvent](#) [RawVideoSampleCapture](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## RawCaptureAsyncEvent

### TVideoGrabber.RawCaptureAsyncEvent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies whether sample captures are returned synchronously or asynchronously.

#### Declaration

**property** RawCaptureAsyncEvent: Boolean **read** GetRawCaptureAsyncEvent **write** SetRawCaptureAsyncEvent **default** DEF\_RawCaptureAsyncEvent;

\_\_property **bool** RawCaptureAsyncEvent=read=GetRawCaptureAsyncEvent, write=SetRawCaptureAsyncEvent, **default**=1

Property RawCaptureAsyncEvent As Boolean

#### Description

Used to specify whether sample captures are returned synchronously or asynchronously by the [OnRawVideoSample](#) or [OnRawAudioSample](#) events.

By default, this property is enabled and the samples are returned asynchronously. It means that samples are captured internally when they travel the video stream and returned later by the events.

If you disable this property, the samples are returned synchronously. This means that the video stream

waits for the capture events to complete.

In this case you must process the samples as fast as possible and you should not perform any actions with the potential to block, such as holding a critical section or waiting on another thread. Also, do not call any GDI or USER32.DLL APIs that might cause a window to move.

#### See Also

[TOnRawSample](#) [OnRawAudioSample](#) [OnRawVideoSample](#) [RawAudioSampleCapture](#)  
[RawVideoSampleCapture](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## RawSampleCaptureLocation

### TVideoGrabber.RawSampleCaptureLocation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the insertion point of the raw sample capture.

#### Declaration

**property** RawSampleCaptureLocation: TRawSampleCaptureLocation **read** GetRawSampleCaptureLocation **write** SetRawSampleCaptureLocation **default** DEF\_RawSampleCaptureLocation;

\_\_property TRawSampleCaptureLocation  
RawSampleCaptureLocation=read=GetRawSampleCaptureLocation, write=SetRawSampleCaptureLocation,  
**default**=0

Property RawSampleCaptureLocation As TRawSampleCaptureLocation

#### Description

This [TRawSampleCaptureLocation](#) property is used to specify the insertion point of the raw sample capture.

nm\_rl\_SourceFormat the sample capture is inserted on the capture device output.

nm\_rl\_AfterCompression the sample capture is inserted after the audio or video compressor, when recording and compressing on the fly.

Created with the Standard Edition of HelpNDoc: [Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion](#)

## RawVideoSampleCapture

### TVideoGrabber.RawVideoSampleCapture

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the capture of the raw video samples.

#### Declaration

**property** RawVideoSampleCapture: Boolean **read** GetRawVideoSampleCapture **write** SetRawVideoSampleCapture **default** DEF\_RawVideoSampleCapture;

\_\_property **bool** RawVideoSampleCapture=read=GetRawVideoSampleCapture,  
write=SetRawVideoSampleCapture, **default**=0

Property RawVideoSampleCapture As Boolean

#### Description

Used to enable / disable the capture of each raw video sample coming out of the video source. Each video sample will be returned by the [OnRawVideoSample](#) event.

## See Also

[TOnRawSample](#) [OnRawAudioSample](#) [OnRawVideoSample](#) [RawAudioSampleCapture](#)  
[RawCaptureAsyncEvent](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## RecordingAudioBitRate

### TVideoGrabber.RecordingAudioBitRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the audio bit rate used for the recording

#### Declaration

RecordingAudioBitRate RecordingTimerInterval: LongInt **read** GetRecordingTimerInterval **write** SetRecordingTimerInterval **default** DEF\_RecordingTimerInterval;

\_\_RecordingAudioBitRate **int** RecordingTimerInterval==GetRecordingTimerInterval,  
write=SetRecordingTimerInterval, **default**=-1;

RecordingAudioBitRate RecordingTimerInterval as Long

#### Description

Used to specify the audio bit rate used for the recording, expressed in Kb/sec.  
Default = -1 (no bit rate specified)

*This value is used by the Datastead RTSP DirectShow source filter.*

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## RecordingBacktimedFramesCount

### TVideoGrabber.RecordingBacktimedFramesCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to record the AVI file starting with a back-timed amount of video frames.

#### Declaration

**property** RecordingBacktimedFramesCount: LongInt **read** GetRecordingBacktimedFramesCount **write** SetRecordingBacktimedFramesCount;

\_\_property **int** RecordingBacktimedFramesCount=read=GetRecordingBacktimedFramesCount,  
write=SetRecordingBacktimedFramesCount, **nodefault**

Property RecordingBacktimedFramesCount As Long

#### Description

Specifies the number of video frames to back-time (preroll) in the buffer.  
E.g. if you want to preroll 3 seconds and the frame rate is 15 fps, set RecordingBacktimedFramesCount = 3 x 15 = 45.  
See the [Back-timed recording \(preroll\)](#) chapter for more information.

## See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#)  
[TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#)  
[AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#)  
[AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#)  
[OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)

[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer](#)

## RecordingCanPause

### TVideoGrabber.RecordingCanPause

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Enables/disables the pause/resume functions during recording.

#### Declaration

**property** RecordingCanPause: Boolean **read** GetRecordingCanPause **write** SetRecordingCanPause;

\_\_property **bool** RecordingCanPause=read=GetRecordingCanPause, write=SetRecordingCanPause, **nodefault**

Property RecordingCanPause As Boolean

#### Description

Used to enable or disable the pause/resume feature during recording.

This property activates the [PauseRecording](#) and [ResumeRecording](#) features during recording.

Disable this property if you get audio / video synchronisation problems.

#### See Also

[Recording methods and properties](#)
[TAVIMuxConfig](#)
[TASFDeinterlaceMode](#)
[TAutoFileName](#)
[TOnRecordingCompleted](#)
[TOnRecordingReadyToStart](#)
[TRecordingMethod](#)
[TSyncPreview](#)
[AudioRecording](#)
[AudioSyncAdjustment](#)
[AutoFileNameMinDigits](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[Encoder](#)
[SetInt](#)
[HoldRecording](#)
[IsRecordingPaused](#)
[Last\\_Recording\\_FileName](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## RecordingDuration

### TVideoGrabber.RecordingDuration

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Duration of last AVI file captured.

**Declaration**

**property** RecordingDuration: Double **read** GetRecordingDuration;

\_\_property **double** RecordingDuration=read=GetRecordingDuration

Property RecordingDuration As Double

**Description**

Retrieves the duration of the last AVI file captured since the application started.

**See Also**

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Streamline your documentation process with HelpNDoc's WinHelp HLP to CHM conversion feature

**RecordingFileName****TVideoGrabber.RecordingFileName**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

File name used to create the recording file.

**Declaration**

**property** RecordingFileName: **string** **read** GetRecordingFileName **write** SetRecordingFileName;

\_\_property wchar\_t \*RecordingFileName=read=GetRecordingFileName, write=SetRecordingFileName

Property RecordingFileName As String

**Description**

Used to set or retrieve the full qualified path and name of the recording file created by [StartRecording](#) . If the file exists, it will be overwritten when the capture starts.

If **RecordingFileName** is left blank, when invoking [StartRecording](#) or [RecordToNewFileNow](#) a file name is generated automatically according to the [StoragePath](#) , [AutoFileName](#) , [AutoFilePrefix](#) and [CaptureFileExt](#) properties.

**See Also**

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#)

[RecordingCanPause](#) [RecordingDuration](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Documents to eBooks: A Step-by-Step Guide with HelpNDoc](#)

## RecordingFileSizeMaxInMB

### TVideoGrabber.RecordingFileSizeMaxInMB

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies a maximum recording file size

#### Declaration

**property** RecordingFileSizeMaxInMB: LongInt **read** GetRecordingFileSizeMaxInMB **write** SetRecordingFileSizeMaxInMB **default** DEF\_RecordingFileSizeMaxInMB;

\_\_**property** Int RecordingFileSizeMaxInMB=read=GetRecordingFileSizeMaxInMB, write=SetRecordingFileSizeMaxInMB, **default**=0 ;

Property RecordingFileSizeMaxInMB As Integer

#### Description

If set to a non-zero value before invoking StartRecording, during the recording, each time the file size being recorded exceeds the specified size expressed in Mb, a new file is generated on the fly, depending on the auto file name settings.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## RecordingFourCC

### TVideoGrabber.RecordingFourCC

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the FourCC used to record the video stream

#### Declaration

**property** RecordingFourCC: **String** **read** GetRecordingFourCC;

\_\_**property** wchar\_t \*RecordingFourCC=read=GetRecordingFourCC



Property RecordingFourCC As String

### Description

Returns the FourCC that corresponds to the video format used to record the video stream.

E.g.:  
YUY2  
RGB24  
dvsd

Note:

When the frame grabber is enabled on the recording stream ([FrameGrabber](#) = fg\_BothStreams or fg\_CaptureStream), the recorded format will be a RGB... format corresponding to the [FrameGrabberRGBFormat](#) property.

To record in the format specified by the [VideoSubtype](#) property, you can

- disable the frame grabber on place it on the preview stream only ( ([FrameGrabber](#) = fg\_Disabled or fg\_PreviewStream),
- force the recording in the native format by enabling the [RecordingInNativeFormat](#) property.

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## RecordingHeight

### TVideoGrabber.RecordingHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the height of video frames in the video stream saved to the AVI file during [recording](#) .

### Declaration

**property** RecordingHeight: LongInt **read** GetRecordingHeight;

\_\_\_property **int** RecordingHeight=read=GetRecordingHeight, **nodefault**

Property RecordingHeight As Long

### Description

Used to retrieve the height of video frames in the video stream saved to the AVI file during [recording](#) .

The returned value is usually the same as [VideoHeight](#) , except when capturing DV in native interleaved mode (when [RecordingInNativeFormat](#) enabled). In this case, the captured size is always the full DV resolution (720x576 in PAL and 720x480 in NTSC), whatever the [video size](#) used for preview.

### See Also



[Recording methods and properties](#)
[TAVIMuxConfig](#)
[TASFDeinterlaceMode](#)
[TAutoFileName](#)
[TOnRecordingCompleted](#)
[TOnRecordingReadyToStart](#)
[TRecordingMethod](#)
[TSyncPreview](#)
[AudioRecording](#)
[AudioSyncAdjustment](#)
[AutoFileNameMinDigits](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[Encoder](#)
[SetInt](#)
[HoldRecording](#)
[IsRecordingPaused](#)
[Last](#)
[Recording](#)
[FileName](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

## RecordingInNativeFormat

### TVideoGrabber.RecordingInNativeFormat

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Properties](#)

Specifies that the video stream must be saved as it comes out of the video device (no overlay, no compression).

#### Declaration

**property** RecordingInNativeFormat: Boolean **read** GetRecordingInNativeFormat **write** SetRecordingInNativeFormat **default** DEF\_RecordingInNativeFormat;

\_\_property **bool** RecordingInNativeFormat=read=GetRecordingInNativeFormat, write=SetRecordingInNativeFormat, **default**=1

Property RecordingInNativeFormat As Boolean

#### Description

Used to specify if the video stream must be saved "as is" into the AVI file, or if it can be modified (compression, frame overlay, etc...)

**enabled (default):** the video stream is saved as it comes out of the video capture device, it cannot be modified,

**disabled:** the video stream is saved after going through the frame grabber (e.g. for text or graphics overlay) and/or a video compressor (for "on the fly" compression).

#### See Also

[Recording methods and properties](#)
[TAVIMuxConfig](#)
[TASFDeinterlaceMode](#)
[TAutoFileName](#)
[TOnRecordingCompleted](#)
[TOnRecordingReadyToStart](#)
[TRecordingMethod](#)
[TSyncPreview](#)
[AudioRecording](#)
[AudioSyncAdjustment](#)
[AutoFileNameMinDigits](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[Encoder](#)
[SetInt](#)
[HoldRecording](#)
[IsRecordingPaused](#)
[Last](#)
[Recording](#)
[FileName](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

## RecordingMethod

### TVideoGrabber.RecordingMethod

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the recording method.

#### Declaration

**property** RecordingMethod: TRecordingMethod **read** GetRecordingMethod **write** SetRecordingMethod  
**default** DEF\_RecordingMethod;

\_\_property TRecordingMethod RecordingMethod=read=GetRecordingMethod, write=SetRecordingMethod, default=0

Property RecordingMethod As TxRecordingMethod

#### Description

This [TRecordingMethod](#) property is used to specify the recording method:

**rm\_AVI** : records the video and audio streams to an AVI file.

**rm\_ASF** : records the video and audio streams to an ASF file.

**rm\_SendToDV** : sends the output to a DV device, like a DV camcorder. See the [Send to DV](#) chapter.

**rm\_MKV**: similar to rm\_AVI, but the recording will create a MKV file (the Matroska muxer must be installed, it can be downloaded at <http://www.matroska.org/>)

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

## RecordingOnMotion\_Enabled

### TVideoGrabber.RecordingOnMotion\_Enabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Activates the recording only when motion is detected.

#### Declaration

**property** RecordingOnMotion\_Enabled: Boolean **read** GetRecordingOnMotion\_Enabled **write** SetRecordingOnMotion\_Enabled **default** DEF\_RecordingOnMotion\_Enabled;

\_\_property **bool** RecordingOnMotion\_Enabled=read=GetRecordingOnMotion\_Enabled,

write=SetRecordingOnMotion\_Enabled, **default**=0

Property RecordingOnMotion\_Enabled As Boolean

### Description

Used to activate the recording only when motion is detected. Then the recording is paused until motion is detected again.

Must be activated before invoking [StartRecording](#).

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_ShowGridDialog MotionDetector\\_Triggered MotionDetector\\_UseThisReferenceSample OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected RecordingOnMotion MotionThreshold RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Import and export Markdown documents

## RecordingOnMotion\_MotionThreshold

### TVideoGrabber.RecordingOnMotion\_MotionThreshold

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Motion threshold that activates the recording.

### Declaration

**property** RecordingOnMotion\_MotionThreshold: Double **read** GetRecordingOnMotion\_MotionThreshold  
**write** SetRecordingOnMotion\_MotionThreshold;

\_\_property **double** RecordingOnMotion\_MotionThreshold=read=GetRecordingOnMotion\_MotionThreshold,  
write=SetRecordingOnMotion\_MotionThreshold

Property RecordingOnMotion\_MotionThreshold As Double

### Description

When [RecordingOnMotion\\_Enabled](#) is activated, this property specifies a minimal motion ratio above which the recording is (re-)activated.

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity](#)

[MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

## RecordingOnMotion\_NoMotionPauseDelayMs

### TVideoGrabber.RecordingOnMotion\_NoMotionPauseDelayMs

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

"no motion" delay after which the recording is paused.

#### Declaration

**property** RecordingOnMotion\_NoMotionPauseDelayMs: LongInt **read** GetRecordingOnMotion\_NoMotionPauseDelayMs **write** SetRecordingOnMotion\_NoMotionPauseDelayMs **default** DEF\_RecordingOnMotion\_NoMotionPauseDelayMs;

\_\_property int

RecordingOnMotion\_NoMotionPauseDelayMs=read=GetRecordingOnMotion\_NoMotionPauseDelayMs, write=SetRecordingOnMotion\_NoMotionPauseDelayMs, **default**=2000

Property RecordingOnMotion\_NoMotionPauseDelayMs As Long

#### Description

When [RecordingOnMotion\\_Enabled](#) is activated and no motion is detected, this property specifies a delay after which the recording is switched back in a paused state, waiting for the next motion to occur. The value is expressed in ms (e.g. 2000 = 2 seconds).

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## RecordingPauseCreatesNewFile

### TVideoGrabber.RecordingPauseCreatesNewFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Generates a new recording file each time the recording is paused.

#### Declaration

**property** RecordingPauseCreatesNewFile: Boolean **read** GetRecordingPauseCreatesNewFile **write** SetRecordingPauseCreatesNewFile **default** DEF\_RecordingOnMotion\_CreateNewFiles;

\_\_property **bool** RecordingPauseCreatesNewFile=read=GetRecordingPauseCreatesNewFile, write=SetRecordingPauseCreatesNewFile, **default**=0

Property RecordingPauseCreatesNewFile As Boolean

### Description

When pausing the recording and this property is enabled, a new recording file name is generated automatically.

The new file is not created when resuming the recording, but rather when the recording is paused by invoking [PauseRecording](#).

By this way, the file is already created and opened when invoking [ResumeRecording](#), so no frames are lost at this time.

If the recording is stopped before invoking [ResumeRecording](#), the new empty file (that has just been created before, when [PauseRecording](#) has been invoked) is deleted.

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Keep Your PDFs Safe from Unauthorized Access with These Security Measures](#)

## RecordingSize

### TVideoGrabber.RecordingSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to record at a smaller size than the preview size

### Declaration

**property** RecordingSize: TRecordingSize **read** GetRecordingSize **write** SetRecordingSize **default** DEF\_RecordingSize;

\_\_property TRecordingSize RecordingSize=read=GetRecordingSize, write=SetRecordingSize, **default**=0 ;

Property RecordingSize As TxRecordingSize

### Description

Used to record in AVI at a smaller size than the preview size.

By default, the recording is made at the same size than the preview size.

Possible values:



**rs\_Default** : records at the current video size  
**rs\_HalfSize** : records at half size  
**rs\_QuarterSize** : records at quarter size

This property lets you record smaller at a smaller size, while keeping an higher resolution for frame capture.

E.g. if the current preview size is 640x480 and RecordingSize = HalfSize, the AVI recording will be mad in 320x240.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## RecordingTimer

### TVideoGrabber.RecordingTimer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to enable the timed recording

#### Declaration

**property** RecordingTimer: TRecordingTimer **read** GetRecordingTimer **write** SetRecordingTimer **default** DEF\_RecordingTimer;

**\_\_property** TRecordingTimer RecordingTimer==GetRecordingTimer, write=SetRecordingTimer, **default**=0;

property RecordingTimer as TxRecordingTimer

#### Description

Used to enable the timed recording:

**rt\_Disabled** : disabled

**rt\_RecordToNewFile** : a new recording file is generated each [RecordingTimerInterval](#) seconds

**rt\_StopRecording** : the recording stops after [RecordingTimerInterval](#) seconds

**rt\_StartRecording** : the recording starts in preview mode only, the recording will begin after [RecordingTimerInterval](#) seconds

**rt\_FrameCapture** : Captures periodically the current frame, the type of capture (memory bitmap, bmp file, etc) is determined by the BurstType property. The interval is determined by RecordingTimerInterval, expressed in seconds

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)

[OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#)  
[OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#)  
[OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#)  
[PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#)  
[RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#)  
[RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#)  
[RecordingMethod](#) [RecordingSize](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#)  
[SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#)  
[StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

## RecordingTimerInterval

### TVideoGrabber.RecordingTimerInterval

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Interval of time for RecordingTimer, in seconds

#### Declaration

**property** RecordingTimerInterval: LongInt **read** GetRecordingTimerInterval **write** SetRecordingTimerInterval **default** DEF\_RecordingTimerInterval;

**\_\_property** int RecordingTimerInterval==GetRecordingTimerInterval, write=SetRecordingTimerInterval, default=0;

property RecordingTimerInterval as Long

#### Description

Specifies the Interval of time between each [RecordingTimer](#) action (generates new file or stops recording), expressed in seconds.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#)  
[TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#)  
[AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#)  
[AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#)  
[OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)  
[OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#)  
[OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#)  
[OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#)  
[PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#)  
[RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#)  
[RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#)  
[RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingWidth](#) [ResumeRecording](#)  
[SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#)  
[StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## RecordingVideoBitRate

### TVideoGrabber.RecordingVideoBitRate

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the video bit rate used for the recording



**Declaration**

RecordingVideoBitRate RecordingTimerInterval: LongInt **read** GetRecordingTimerInterval **write** SetRecordingTimerInterval **default** DEF\_RecordingTimerInterval;

\_\_RecordingVideoBitRate **int** RecordingTimerInterval==GetRecordingTimerInterval, write=SetRecordingTimerInterval, **default**=-1;

RecordingVideoBitRate RecordingTimerInterval as Long

**Description**

Used to specify the video bit rate used for the recording, expressed in Kb/sec.  
Default = -1 (no bit rate specified)

**Deprecated**

Created with the Standard Edition of HelpNDoc: [Effortlessly create a professional-quality documentation website with HelpNDoc](#)

**RecordingWidth****TVideoGrabber.RecordingWidth**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the width of video frames in the video stream saved to the AVI file during [recording](#) .

**Declaration**

**property** RecordingWidth: LongInt **read** GetRecordingWidth;

\_\_**property int** RecordingWidth=read=GetRecordingWidth, **nodefault**

Property RecordingWidth As Long

**Description**

Used to retrieve the width of video frames in the video stream saved to the AVI file during [recording](#) .

The returned value is usually the same as [VideoWidth](#) , except when capturing DV in native interleaved mode (when [RecordingInNativeFormat](#) is true). In this case, the captured size is always the full DV resolution (720x576 in PAL and 720x480 in NTSC), whatever the [video size](#) used for preview.

**See Also**

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

**Reencoding\_IncludeAudioStream****TVideoGrabber.Reencoding\_IncludeAudioStream**[Prev](#)[Next](#)

## [TVideoGrabber](#) [Properties](#)

Specifies if the audio stream must be included.

### Declaration

**property** Reencoding\_IncludeAudioStream: Boolean **read** GetReencodingIncludeAudioStream **write** SetReencodingIncludeAudioStream **default** DEF\_Reencoding\_IncludeAudioStream;

\_\_property **bool** Reencoding\_IncludeAudioStream=read=GetReencodingIncludeAudioStream, write=SetReencodingIncludeAudioStream, **default**=1

Property Reencoding\_IncludeAudioStream As Boolean

### Description

Used to specify if the audio stream must be included in the reencoded video clip. If false, the new video clip will contain only the video stream.

### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## Reencoding\_IncludeVideoStream

### [TVideoGrabber.Reencoding\\_IncludeVideoStream](#)

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Specifies if the video stream must be included in the reencoded video clip.

### Declaration

**property** Reencoding\_IncludeVideoStream: Boolean **read** GetReencodingIncludeVideoStream **write** SetReencodingIncludeVideoStream **default** DEF\_Reencoding\_IncludeVideoStream;

\_\_property **bool** Reencoding\_IncludeVideoStream=read=GetReencodingIncludeVideoStream, write=SetReencodingIncludeVideoStream, **default**=1

Property Reencoding\_IncludeVideoStream As Boolean

### Description

Used to specify if the video stream must be included in the reencoded video clip. If false, the new video clip will contain only the audio stream.

### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

## Reencoding\_Method

### [TVideoGrabber.Reencoding\\_Method](#)

[Prev](#)

[Next](#)

## **TVideoGrabber** [Properties](#)

Specifies the reencoding format: AVI or ASF

### Declaration

**property** Reencoding\_Method: TRecordingMethod **read** GetReencodingMethod **write** SetReencodingMethod **default** DEF\_Reencoding\_Method;

\_\_property TRecordingMethod Reencoding\_Method=read=GetReencodingMethod, write=SetReencodingMethod, **default**=0

Property Reencoding\_Method As TRecordingMethod

### Description

Used to specify the reencoding format:

AVI = rm\_AVI

ASF = rm\_ASF

(other [TRecordingMethod](#) values are not supported for this property).

### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## **Reencoding\_NewVideoClip**

### **TVideoGrabber.Reencoding\_NewVideoClip**

[Prev](#)

[Next](#)

#### **TVideoGrabber** [Properties](#)

Specifies the file name of the new video clip.

### Declaration

**property** Reencoding\_NewVideoClip: **string** **read** GetReencodingNewVideoClip **write** SetReencodingNewVideoClip;

\_\_property wchar\_t \*Reencoding\_NewVideoClip=read=GetReencodingNewVideoClip, write=SetReencodingNewVideoClip

Property Reencoding\_NewVideoClip As String

### Description

Used to specify the file name of the video clip that will be created when invoking [StartReencoding](#).

### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## Reencoding\_SourceVideoClip

### TVideoGrabber.Reencoding\_SourceVideoClip

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

File name of the video clip to reencode.

#### Declaration

**property** Reencoding\_SourceVideoClip: **String** **read** GetReencodingSourceVideoClip **write** SetReencodingSourceVideoClip;

\_\_property wchar\_t \*Reencoding\_SourceVideoClip=read=GetReencodingSourceVideoClip, write=SetReencodingSourceVideoClip

Property Reencoding\_SourceVideoClip As String

#### Description

Used to specify the file name of the video clip that will be reencoded when invoking [StartReencoding](#).

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion](#)

## Reencoding\_StartFrame

### TVideoGrabber.Reencoding\_StartFrame

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Specifies the start frame if the video clip must be cut.

#### Declaration

**property** Reencoding\_StartFrame: LargeInteger **read** GetReencodingStartFrame **write** SetReencodingStartFrame **default** -1;

\_\_property \_\_int64 Reencoding\_StartFrame=read=GetReencodingStartFrame, write=SetReencodingStartFrame, **default**=-1

Property Reencoding\_StartFrame as Double

#### Description

Used to specify the start frame if the new video clip must contain only a part of the original video clip. Set the value to -1 to start from the beginning of the video clip.

Use [AVIDuration](#) to retrieve the number of frames of the video clip (before invoking [StartReencoding](#)).

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP](#)

[to CHM conversion tool](#)

## Reencoding\_StartTime

### TVideoGrabber.Reencoding\_StartTime

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the start time if the video clip must be cut.

#### Declaration

**property** Reencoding\_StartTime: LargeInteger **read** GetReencodingStartTime **write** SetReencodingStartTime **default** DEF\_Reencoding\_StartTime;

\_\_property \_\_int64 Reencoding\_StartTime=read=GetReencodingStartTime, write=SetReencodingStartTime, default=- 1

Property Reencoding\_StartTime As Double

#### Description

Used to specify the start time if the new video clip must contain only a part of the original video clip, expressed in 100ns units (e.g. 4 seconds = 40000000).

Set the value to **-1** (or **0**) to start from the beginning of the video clip.

Use [AVIDuration](#) to retrieve the duration of the video clip (before invoking [StartReencoding](#)).

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## Reencoding\_StopFrame

### TVideoGrabber.Reencoding\_StopFrame

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the stop frame if the video clip must be cut.

#### Declaration

**property** Reencoding\_StopFrame: LargeInteger **read** GetReencodingStopFrame **write** SetReencodingStopFrame **default** -1;

\_\_property \_\_int64 Reencoding\_StopFrame=read=GetReencodingStopFrame, write=SetReencodingStopFrame, default=-1

Property Reencoding\_StopFrame as Double

#### Description

Used to specify the stop frame if the new video clip must contain only a part of the original video clip.

Set the value to **-1** to go on until the end of the video clip.

Use [AVIDuration](#) to retrieve the frame count of the video clip (before invoking [StartReencoding](#)).

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopTime](#)

[Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#)  
[Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## Reencoding\_StopTime

### TVideoGrabber.Reencoding\_StopTime

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the stop time if the video clip must be cut.

#### Declaration

**property** Reencoding\_StopTime: LargeInteger **read** GetReencodingStopTime **write** SetReencodingStopTime **default** DEF\_Reencoding\_StopTime;

\_\_property \_\_int64 Reencoding\_StopTime=read=GetReencodingStopTime, write=SetReencodingStopTime, default=- 1

Property Reencoding\_StopTime As Double

#### Description

Used to specify the stop time if the new video clip must contain only a part of the original video clip, expressed in 100ns units (e.g. 4 seconds = 40000000).

Set the value to **-1** to end at the normal duration of the video clip.

Use [AVIDuration](#) to retrieve the duration of the video clip (before invoking [StartReencoding](#)).

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#)  
[Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#)  
[Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#)  
[Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#)  
[Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## Reencoding\_UseAudioCompressor

### TVideoGrabber.Reencoding\_UseAudioCompressor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the current audio compressor to use.

#### Declaration

**property** Reencoding\_UseAudioCompressor: Boolean **read** GetReencodingUseAudioCompressor **write** SetReencodingUseAudioCompressor **default** DEF\_Reencoding\_UseAudioCompressor;

\_\_property **bool** Reencoding\_UseAudioCompressor=read=GetReencodingUseAudioCompressor, write=SetReencodingUseAudioCompressor, **default**=0

Property Reencoding\_UseAudioCompressor As Boolean

#### Description

Used to specify if the current [audio compressor](#) must to use to reencode the audio stream of the video clip.

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#)

[Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#)  
[Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#)  
[Reencoding\\_StopTime](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#)  
[Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

## Reencoding\_UseFrameGrabber

### TVideoGrabber.Reencoding\_UseFrameGrabber

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies if the frame grabber must be used when reencoding.

#### Declaration

**property** Reencoding\_UseFrameGrabber: Boolean **read** GetReencodingUseFrameGrabber **write** SetReencodingUseFrameGrabber **default** DEF\_Reencoding\_UseFrameGrabber;

\_\_property **bool** Reencoding\_UseFrameGrabber=read=GetReencodingUseFrameGrabber, write=SetReencodingUseFrameGrabber, **default**=1

Property Reencoding\_UseFrameGrabber As Boolean

#### Description

Used to specify if the [frame grabber](#) must to use to reencode the video stream of the video clip. Using the frame grabber lets you apply text and/or graphics overlay over the video frames.

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#)  
[Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#)  
[Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#)  
[Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseVideoCompressor](#)  
[Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## Reencoding\_UseVideoCompressor

### TVideoGrabber.Reencoding\_UseVideoCompressor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the current video compressor to use.

#### Declaration

**property** Reencoding\_UseVideoCompressor: Boolean **read** GetReencodingUseVideoCompressor **write** SetReencodingUseVideoCompressor **default** DEF\_Reencoding\_UseVideoCompressor;

\_\_property **bool** Reencoding\_UseVideoCompressor=read=GetReencodingUseVideoCompressor, write=SetReencodingUseVideoCompressor, **default**=0

Property Reencoding\_UseVideoCompressor As Boolean

#### Description

Used to specify if the current [video compressor](#) must to use to reencode the video stream of the video clip.

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#)  
[Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#)



[Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

## Reencoding\_WMVOutput

### TVideoGrabber.Reencoding\_WMVOutput

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies if the reencoding output must be in WMV format

#### Declaration

**property** Reencoding\_WMVOutput: Boolean **read** GetReencodingWMVOutput **write** SetReencodingWMVOutput **default** DEF\_Reencoding\_WMVOutput;

\_\_property **bool** Reencoding\_WMVOutput=read=GetReencodingWMVOutput, write=SetReencodingWMVOutput, **default**=1

Property Reencoding\_WMVOutput As Boolean

#### Description

Used to specify if the reencoding output must be in WMV format.

#### 1) [Reencoding\\_WMVOutput](#) enabled:

the extension of the video clip specified in [Reencoding\\_NewVideoClip](#) will be ".wmv".

#### 1) [Reencoding\\_WMVOutput](#) disabled:

you have to specify the proper value according to the reencoding settings:

- a) if [Reencoding\\_Method](#) = rm\_ASF, the extension must be **asf**,
- b) if [Reencoding\\_Method](#) = rm\_AVI, the extension must be:
  - **mpg** if the video clip will contain MPEG video,
  - **avi** if the video clip will contain only video, or both audio and video, compressed or not
  - **wav** if the video clip contains only uncompressed audio
  - **mp3** if the video clip contains only mp3 audio.

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## ScreenRecordingLayeredWindows

### TVideoGrabber.ScreenRecordingLayeredWindows

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Activates the screen recording of layered windows

#### Declaration

**property** ScreenRecordingLayeredWindows: Boolean **read** GetScreenRecordingLayeredWindows **write** SetScreenRecordingLayeredWindows **default** DEF\_ScreenRecordingLayeredWindows;

**\_\_property bool** ScreenRecordingLayeredWindows=read=GetScreenRecordingLayeredWindows, write=SetScreenRecordingLayeredWindows, **default=0**

Property ScreenRecordingLayeredWindows as Boolean

#### Description

Used to enable the screen recording of layered windows (e.g. floating or transparent windows).

By default the layered windows are not captured, enabling this property activates it.

Note that enabling this property may can affect the screen recording performances and/or cause a blinking cursor symptom.

#### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Transform your help documentation into a stunning website](#)

## ScreenRecordingMonitor

### TVideoGrabber.ScreenRecordingMonitor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of the monitor to record when performing screen recording.

#### Declaration

**property** ScreenRecordingMonitor: LongInt **read** GetScreenRecordingMonitor **write** SetScreenRecordingMonitor **default** 0

**\_\_property int** ScreenRecordingMonitor==GetScreenRecordingMonitor, write=SetScreenRecordingMonitor, **default=0**;

Property ScreenRecordingMonitor as Long

#### Description

Specifies the monitor to record when using the screen recording and the desktop is extended to several monitors.

Just specify the monitor number (0 for the 1st monitor, 1 for the 2nd monitor, etc)...

#### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

## ScreenRecordingNonVisibleWindows

### TVideoGrabber.ScreenRecordingNonVisibleWindows

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Activates the screen recording of non-visible windows

#### Declaration

**property** ScreenRecordingNonVisibleWindows: Boolean **read** GetScreenRecordingNonVisibleWindows **write** SetScreenRecordingNonVisibleWindows **default** DEF\_ScreenRecordingNonVisibleWindows;

\_\_**property bool** ScreenRecordingNonVisibleWindows=read=GetScreenRecordingNonVisibleWindows, write=SetScreenRecordingNonVisibleWindows, **default**=0

Property ScreenRecordingNonVisibleWindows as Boolean

### Description

Used to enable the screen recording of non-visible windows.

The window must have been selected first with [SetWindowRecordingByName](#) or

[SetWindowRecordingByHandle](#)

*Note: this mode may give variable results depending on the content of the window*

### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

## ScreenRecordingSizePercent

### TVideoGrabber.ScreenRecordingSizePercent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Downsizes the video size of the screen recording

### Declaration

**property** ScreenRecordingSizePercent: LongInt **read** GetScreenRecordingSizePercent **write** SetScreenRecordingSizePercent **default** 100

\_\_**property int** ScreenRecordingSizePercent==ScreenRecordingSizePercent, write=SetScreenRecordingSizePercent, **default**=100;

Property ScreenRecordingSizePercent as Long

### Description

By default, when ScreenRecordingSizePercent = 100, the video size of the screen recording is the pixel size of the monitor (100%)

To preview or record e.g. at half of the monitor size (50%), set ScreenRecordingSizePercent = 50

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## ScreenRecordingThroughClipboard

### TVideoGrabber.ScreenRecordingThroughClipboard

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Uses the clipboard to perform the screen recording.

### Declaration

**property** ScreenRecordingThroughClipboard: Boolean **read** GetScreenRecordingThroughClipboard **write** SetScreenRecordingThroughClipboard **default** DEF\_ScreenRecordingThroughClipboard;

\_\_**property bool** ScreenRecordingThroughClipboard=read=GetScreenRecordingThroughClipboard, write=SetScreenRecordingThroughClipboard, **default**=0

Property ScreenRecordingThroughClipboard As Boolean

#### Description

When enabled, the clipboard is used to perform the screen recording.

Enabling this property may let you capture more information than when it is disabled (by default).

If some parts of the screen are not properly captured or recorded, try to switch the value of this property.

Note: any existing clipboard data will be destroyed during preview or recording when this property is enabled.

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

### ScreenRecordingWithCursor

#### TVideoGrabber.ScreenRecordingWithCursor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Shows the cursor during the screen recording.

#### Declaration

**property** ScreenRecordingWithCursor: Boolean **read** GetScreenRecordingWithCursor **write** SetScreenRecordingWithCursor **default** DEF\_ScreenRecordingWithCursor;

\_\_property **bool** ScreenRecordingWithCursor=read=GetScreenRecordingWithCursor, write=SetScreenRecordingWithCursor, **default**=1

Property ScreenRecordingWithCursor As Boolean

#### Description

Enables the cursor recording during the screen recording.  
When disabled, the cursor does not appear.

#### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)

### SendToDV\_DeviceIndex

#### TVideoGrabber.SendToDV\_DeviceIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Index of the DV device that will receive the audio/video streams.

#### Declaration

**property** SendToDV\_DeviceIndex: LongInt **read** GetSendToDV\_DeviceIndex **write** SetSendToDV\_DeviceIndex;

\_\_property **int** SendToDV\_DeviceIndex=read=GetSendToDV\_DeviceIndex, write=SetSendToDV\_DeviceIndex, **nodefault**

Property SendToDV\_DeviceIndex As Long

## Description

This property is used to specify the index of the DV device (in the [VideoDevices](#) list) that will receive the audio/video streams when the [RecordingMethod](#) is set to **rm\_SendToDV**. See the "[Send to DV](#)" chapter for more information.

## See Also

[Send to DV](#) [DVEncoder](#) [VideoFormat](#) [DVEncoder](#) [VideoResolution](#) [DVEncoder](#) [VideoStandard](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## ShapeOverlay

### TVideoGrabber.ShapeOverlay

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

TShape component that will be drawn over video frames.

## Declaration

**property** ShapeOverlay: TShape **read** GetShapeOverlay **write** SetShapeOverlay **default** DEF\_ShapeOverlay;

\_\_property ExtCtrls::TShape \*ShapeOverlay=read=GetShapeOverlay, write=SetShapeOverlay, **default**=0

n/a

## Description

Used to assign to TVideoGrabber a TShape component that will be drawn over video frames. The TShape component is really drawn over video frames when [ShapeOverlayEnabled](#) is true.

To use this feature, simply put a TShape component on the form, then, using the object inspector, assign ShapeOverlay to it.

It is possible to draw **more than one shape** over video frames by using [ShapeOverlayList](#) instead of this property.

## Notes:

- to avoid having TShape components placed on a form visible at runtime, simply set their Visible property to false.
- to use this property, the [frame grabber](#) is enabled.

## See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)

[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlayEnabled](#)  
[ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#)  
[TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#)  
[TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#)  
[TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#)  
[TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## ShapeOverlayEnabled

### TVideoGrabber.ShapeOverlayEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables drawing of the [ShapeOverlay](#) TShape component over video frames.

#### Declaration

**property** ShapeOverlayEnabled: Boolean **read** GetShapeOverlayEnabled **write** SetShapeOverlayEnabled **default** DEF\_ShapeOverlayEnabled;

**\_\_property** **bool** ShapeOverlayEnabled=read=GetShapeOverlayEnabled, write=SetShapeOverlayEnabled, **default**=0

n/a

#### Description

Used to enable/disable drawing of the [ShapeOverlay](#) TShape component over video frames. Available only if the [frame grabber](#) is enabled.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#)  
[TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#)  
[TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#)  
[TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#)  
[TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Elevate your documentation to new heights with HelpNDoc's built-in SEO](#)



## SourceStream

### TVideoGrabber.SourceStream

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Used to play a video clip from a TStream

#### Declaration

**property** SourceStream: TStream **read** GetSourceStream **write** SetSourceStream;

\_\_property **TStream** \*SourceStream=read=GetSourceStream, write=SetSourceStream

#### Description

Used to play a video clip from a TMemoryStream, TFileStream or TStream descendent.

When this property is assigned before invoking [OpenPlayer](#), the stream data is used as source by the player.

To go back to the normal player behavior, set a NULL or **nil** value to this property after invoking [ClosePlayer](#) to let TVideoGrabber use the clip specified in the [PlayerFileName](#) property.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## SpeakerBalance

### TVideoGrabber.SpeakerBalance

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Sets the speaker balance.

#### Declaration

**property** SpeakerBalance: LongInt **read** GetSpeakerBalance **write** SetSpeakerBalance;

\_\_property **int** SpeakerBalance=read=GetSpeakerBalance, write=SetSpeakerBalance, **nodefault**

Property SpeakerBalance As Long

#### Description

Used to set the speaker balance. This property is active when [SpeakerControl](#) is enabled.

Valid range: -32768 ... 32767.

0 is the center point.

If [AssociateAudioAndVideoDevices](#) is enabled, the current value is saved with the current [video capture device](#), and automatically restored then this video capture device is selected again (when [VideoDevice](#) is assigned).



#### See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## SpeakerControl

### TVideoGrabber.SpeakerControl

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the control of the speaker balance and volume.

#### Declaration

**property** SpeakerControl: Boolean **read** GetSpeakerControl **write** SetSpeakerControl **default** DEF\_SpeakerControl;

\_\_property **bool** SpeakerControl=read=GetSpeakerControl, write=SetSpeakerControl, **default**=0

Property SpeakerControl As Boolean

#### Description

Used to enable/disable the control of the speaker balance ([SpeakerBalance](#)) and volume ([SpeakerVolume](#)) from TVideoGrabber.

#### See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerBalance](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## SpeakerVolume

### TVideoGrabber.SpeakerVolume

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets the speaker volume.

#### Declaration

**property** SpeakerVolume: LongInt **read** GetSpeakerVolume **write** SetSpeakerVolume;

\_\_property **int** SpeakerVolume=read=GetSpeakerVolume, write=SetSpeakerVolume, **nodefault**

Property SpeakerVolume As Long

#### Description

Used to set the speaker volume. This property is active when [SpeakerControl](#) is enabled. Valid range: 0...65535.

If [AssociateAudioAndVideoDevices](#) is enabled, the current value is saved with the current [video capture device](#), and automatically restored then this video capture device is selected again (when [VideoDevice](#) is assigned).

#### See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerBalance](#) [SpeakerControl](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Make the switch to CHM with HelpNDoc's hassle-free WinHelp HLP to CHM conversion tool](#)

## StoragePath

### TVideoGrabber.StoragePath

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Sets or retrieves the full qualified path for recordings and BMP or JPEG files. This property is NOT stored.

#### Declaration

**property** StoragePath: **string** **read** GetStoragePath **write** SetStoragePath **stored** False;

\_\_property wchar\_t \*StoragePath=read=GetStoragePath, write=SetStoragePath, stored=false

Property StoragePath As String

#### Description

Used to set or retrieve the full qualified path where recordings and BMP or JPEG frame captures will be saved.

If left blank, the current directory is used.

As this property depends of the current platform and partition layout, it is not stored and must be set programmatically at runtime.

**WARNING:** be careful not to use a slow drive as storage path (networked drive, floppy disk, slow hard disk, etc...), otherwise it will slow down recording and cause an number of dropped frames.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [Encoder](#) [SetInt](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [HoldRecording](#) [IsRecordingPaused](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnFrameBitmap](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetFrameCaptureBounds](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## StoreDeviceSettingsInRegistry

### TVideoGrabber.StoreDeviceSettingsInRegistry

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

If true, device-dependent settings are saved in the registry.

#### Declaration

**property** StoreDeviceSettingsInRegistry: Boolean **read** GetStoreDeviceSettingsInRegistry **write** SetStoreDeviceSettingsInRegistry **default** DEF\_StoreDeviceSettingsInRegistry;

\_\_property **bool** StoreDeviceSettingsInRegistry=read=GetStoreDeviceSettingsInRegistry, write=SetStoreDeviceSettingsInRegistry, **default**=1

Property StoreDeviceSettingsInRegistry As Boolean

#### Description

Used to save device-dependent settings in the registry.

If enabled, TVideoGrabber automatically saves and retrieves the current value of all the [device-dependent](#) parameters whose value and range change from a video capture device to the other.

#### See Also

[FindIndexInListByName](#) [OnVideoDeviceSelected](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)

## StreamingURL

### TVideoGrabber.StreamingURL

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

URL that must be given to users to connect to the current live streaming.

#### Declaration

**property** StreamingURL: **string** **read** GetStreamingURL;

\_\_property wchar\_t \*StreamingURL=read=GetStreamingURL

Property StreamingURL As String

#### Description

Returns the URL that must be given to users to connect to the current direct network streaming.

This URL is also returned by the [OnDirectNetworkStreamingHostUrl](#) event when starting direct network streaming.

E.g.:

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#)

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## StreamInterface\_Format

### TVideoGrabber.StreamInterface\_Format

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the format of the raw samples (e.g. H264) pushed through the [StreamInterface.PushData](#) function

#### Declaration

**property** StreamInterface\_Format: **string read** GetStreamInterface\_Format **write** SetStreamInterface\_Format;

\_\_**property** System::wchar\_t \*StreamInterface\_Format==GetStreamInterface\_Format, write=SetStreamInterface\_Format;

### Description

Specifies the format of the raw samples (e.g. H264) pushed through the [StreamInterface\\_PushData](#) function

### See Also

[Stream Interface](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

## StreamInterface\_FrameRate

### TVideoGrabber.StreamInterface\_FrameRate

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the frame rate of the raw samples (e.g. H264) pushed through the [StreamInterface\\_PushData](#) function

### Declaration

**property** StreamInterface\_FrameRate: Double **read** GetStreamInterface\_FrameRate **write** SetStreamInterface\_FrameRate;

\_\_**property double** StreamInterface\_FrameRate==GetStreamInterface\_FrameRate, write=SetStreamInterface\_FrameRate;

### Description

Specifies the frame rate of the raw samples (e.g. H264) pushed through the [StreamInterface\\_PushData](#) function

### See Also

[Stream Interface](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Word Doc into a Professional-Quality eBook with HelpNDoc](#)

## StreamInterface\_IsRealTime

### TVideoGrabber.StreamInterface\_IsRealTime

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies whether the raw samples (e.g. H264) pushed through the [StreamInterface\\_PushData](#) function are samples received in real-time, or samples coming e.g. from a file

### Declaration

**property** StreamInterface\_IsRealTime: Boolean **read** GetStreamInterface\_IsRealTime **write** SetStreamInterface\_IsRealTime;

\_\_**property bool** StreamInterface\_IsRealTime==GetStreamInterface\_IsRealTime, write=SetStreamInterface\_IsRealTime, **nodefault**};

### Description

Specifies whether the raw samples (e.g. H264) pushed through the [StreamInterface\\_PushData](#) function are samples received in real-time, or samples coming e.g. from a file

**See Also**  
[stream interface](#)

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## SyncCommands

### TVideoGrabber.SyncCommands

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Synchronize the TVideoGrabber commands.

#### Declaration

**property** SyncCommands: Boolean **read** GetSyncCommands **write** SetSyncCommands **default** DEF\_SyncCommands;

\_\_property **bool** SyncCommands=read=GetSyncCommands, write=SetSyncCommands, **default**=0

property SyncCommands as Boolean

#### Description

By default this property is enabled so TVideoGrabber is blocking and synchronone (it waits for the commands to be completed)

If you want to invoke the component from a thread to avoid blocking the main thread, disable this property.

See also [EventNotificationSynchronone](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## SynchronizationRole

### TVideoGrabber.SynchronizationRole

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specify if the synchronized player is a master or slave

#### Declaration

**property** SynchronizationRole: TSynchronizationRole **read** GetSynchronizationRole **write** SetSynchronizationRole **default** DEF\_SynchronizationRole;

\_\_property TSynchronizationRole SynchronizationRole=read=GetSynchronizationRole, write=SetSynchronizationRole, **default**=0

Property SynchronizationRole as TSynchronizationRole

#### Description

Used when the players are synchronized ([Synchronized](#) = true), to determine if the player is a master player or a slave player.

sr\_Master: this player is the master component

sr\_Slave: this player is a slave component

(only one master player allowed at the time)

**See Also**

[Player features](#)
[TOnPlayerStateChanged](#)
[TPlayerState](#)
[TOnPlayerBufferingData](#)
[AudioChannelRenderMode](#)
[AudioStreamNumber](#)
[AutoStartPlayer](#)
[AVIDuration](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[AVIInfo2](#)
[ClosePlayer](#)
[FastForwardPlayer](#)
[IsPlayerAudioStreamAvailable](#)
[IsPlayerVideoStreamAvailable](#)
[Last\\_Clip\\_Played](#)
[MP4NeedsReindexing](#)
[OnPlayerBufferingData](#)
[OnPlayerEndOfStream](#)
[OnPlayerStateChanged](#)
[OnPlayerUpdateTrackbarPosition](#)
[OpenPlayer](#)
[OpenPlayerAtFramePositions](#)
[OpenPlayerAtTimePositions](#)
[PausePlayer](#)
[PlayerAudioCodec](#)
[PlayerAudioRendering](#)
[PlayerDuration](#)
[PlayerDVSize](#)
[PlayerFastSeekSpeedRatio](#)
[PlayerFileName](#)
[PlayerForcedCodec](#)
[PlayerFrameCount](#)
[PlayerFramePosition](#)
[PlayerFrameRate](#)
[PlayerFrameStep](#)
[PlayerRefreshPausedDisplay](#)
[PlayerRefreshPausedDisplayFrameRate](#)
[PlayerSpeedRatio](#)
[PlayerTimePosition](#)
[PlayerTrackBar](#)
[PlayerTrackBarScale](#)
[PlayerTrackBarSynchronone](#)
[PlayerVideoCodec](#)
[RewindPlayer](#)
[RunPlayer](#)
[RunPlayerBackwards](#)
[ShowDialog](#)
[SourceStream](#)
[StopPlayer](#)
[Synchronized](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoPlayableWhileRecording](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Transform Your Help Documentation Process with a Help Authoring Tool

## Synchronized

### TVideoGrabber.Synchronized

[Prev](#)
[Next](#)

**TVideoGrabber** [Properties](#)

Enable the synchronization between several TVideoGrabber components.

#### Declaration

**property** Synchronized: Boolean **read** GetSynchronized **write** SetSynchronized **default** DEF\_Synchronized;

\_\_property **bool** Synchronized=read=GetSynchronized, write=SetSynchronized, **default**=0

Property Synchronized As Boolean

#### Description

Used to enable the synchronization between several TVideoGrabber player components.

See the [Synchronization of several TVideoGrabber components](#) chapter.

#### See Also

[Player features](#)
[Recording methods and properties](#)
[TAVIMuxConfig](#)
[TOnPlayerStateChanged](#)
[TPlayerState](#)
[TASFDeinterlaceMode](#)
[TAutoFileName](#)
[TOnFrameProgress](#)
[TOnPlayerBufferingData](#)
[TOnRecordingCompleted](#)
[TOnRecordingReadyToStart](#)
[TOnResizeVideo](#)
[TRecordingMethod](#)
[TSyncPreview](#)
[AudioChannelRenderMode](#)
[AudioRecording](#)
[AudioStreamNumber](#)
[AudioSyncAdjustment](#)
[AutoFileNameMinDigits](#)
[AutoRefreshPreview](#)
[AutoStartPlayer](#)
[AVIDuration](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[AVIInfo2](#)
[BackgroundColor](#)
[ClosePlayer](#)
[Display\\_AutoSize](#)
[Display\\_FullScreen](#)
[Display\\_SetLocation](#)
[Display\\_VideoPortEnabled](#)
[Encoder\\_SetInt](#)
[FastForwardPlayer](#)
[HoldRecording](#)
[ImageRatio](#)
[InFrameProgressEvent](#)
[IsPlayerAudioStreamAvailable](#)
[IsPlayerVideoStreamAvailable](#)
[IsRecordingPaused](#)
[IsVideoInterlaced](#)
[IsVideoPortAvailable](#)
[Last\\_Clip\\_Played](#)
[Last\\_Recording\\_FileName](#)
[MP4NeedsReindexing](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnFrameProgress](#)
[OnKeyPress](#)
[OnPlayerBufferingData](#)
[OnPlayerEndOfStream](#)
[OnPlayerStateChanged](#)
[OnPlayerUpdateTrackbarPosition](#)
[OnPreviewStarted](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[OnResizeVideo](#)
[OpenPlayer](#)
[OpenPlayerAtFramePositions](#)
[OpenPlayerAtTimePositions](#)
[PausePlayer](#)
[PausePreview](#)
[PauseRecording](#)
[PlayerAudioCodec](#)
[PlayerAudioRendering](#)
[PlayerDuration](#)
[PlayerDVSize](#)
[PlayerFastSeekSpeedRatio](#)
[PlayerFileName](#)
[PlayerForcedCodec](#)
[PlayerFrameCount](#)
[PlayerFramePosition](#)
[PlayerFrameRate](#)
[PlayerFrameStep](#)
[PlayerRefreshPausedDisplay](#)
[PlayerRefreshPausedDisplayFrameRate](#)
[PlayerSpeedRatio](#)
[PlayerTimePosition](#)
[PlayerTrackBar](#)
[PlayerTrackBarScale](#)
[PlayerTrackBarSynchronone](#)
[PlayerVideoCodec](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[PreviewZoomSize](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)



[RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumePreview](#) [ResumeRecording](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [ShowDialog](#) [SourceStream](#) [StartAudioRecording](#) [StartPreview](#) [StartRecording](#) [StartSynchronized](#) [StopPlayer](#) [StopPreview](#) [StopRecording](#) [StoragePath](#) [SynchronizationRole](#) [SyncPreview](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [VideoWidth](#) [PreferredAspectRatio](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## SyncPreview

### TVideoGrabber.SyncPreview

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables the audio/video sync of the preview.

#### Declaration

**property** SyncPreview: TSyncPreview **read** GetSyncPreview **write** SetSyncPreview **default** DEF\_SyncPreview;

\_\_property TSyncPreview SyncPreview=read=GetSyncPreview, write=SetSyncPreview, **default**=0

Property SyncPreview As TSyncPreview

#### Description

This [TSyncPreview](#) property is used to maintain the audio rendered and the video displayed in sync, during preview as well as during recording.

When this property is enabled during recording, this can cause an excessive amount of dropped frames with some video capture devices.

Disabling this property can cause a time shift between the audio rendered and the video displayed.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## SystemTempPath

### TVideoGrabber.SystemTempPath

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns the current system temporary path.



## Declaration

**property** SystemTempPath: **string read** GetSystemTempPath;

\_\_property wchar\_t \*SystemTempPath=read=GetSystemTempPath

Property SystemTempPath As String

## Description

Used to retrieve the current system temporary path.

Can be used e.g. to be assigned to the [StoragePath](#) property, in order to save the recorded clips and captured frames in the temporary directory.

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## TextOverlay\_Align

### TVideoGrabber.TextOverlay\_Align

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the alignment used to draw text over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

## Declaration

**property** TextOverlay\_Align: TTextOverlayAlign **read** GetTextOverlayAlign **write** SetTextOverlayAlign **default** DEF\_TextOverlay\_Align;

\_\_property TTextOverlayAlign TextOverlay\_Align=read=GetTextOverlayAlign, write=SetTextOverlayAlign, default=0

Property TextOverlay\_Align As TTextOverlayAlign

## Description

Used to set or retrieve the alignment used to draw text over video frames.

The text will be drawn between [TextOverlay\\_Left](#) and [TextOverlay\\_Right](#) positions.

## See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay\\_StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#)

[TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Elevate your documentation to new heights with HelpNDoc's built-in SEO](#)

## TextOverlay\_AlphaBlend

### TVideoGrabber.TextOverlay\_AlphaBlend

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the alpha blending of the current text overlay

#### Declaration

**property** TextOverlay\_AlphaBlend: Boolean **read** GetTextOverlay\_AlphaBlend **write** SetTextOverlay\_AlphaBlend **default** DEF\_TextOverlay\_AlphaBlend;

**\_\_property bool** TextOverlay\_AlphaBlend==GetTextOverlay\_AlphaBlend, write=SetTextOverlay\_AlphaBlend, **default**=0;

property TextOverlay\_AlphaBlend as Boolean

#### Description

Enables the alpha blending of an overlayed text selected by [TextOverlay\\_Selector](#)  
The alpha blending value in the 0..255 range must be specified with [TextOverlay\\_AlphaBlendValue](#)  
Look at the [Text Overlays](#) chapter for more information.

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## TextOverlay\_AlphaBlendValue

### TVideoGrabber.TextOverlay\_AlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the the alpha blending value of the current text overlay

#### Declaration

**property** TextOverlay\_AlphaBlendValue: LongInt **read** GetTextOverlay\_AlphaBlendValue **write** SetTextOverlay\_AlphaBlendValue **default** DEF\_TextOverlay\_AlphaBlendValue;

**\_\_property int** TextOverlay\_AlphaBlendValue==GetTextOverlay\_AlphaBlendValue, write=SetTextOverlay\_AlphaBlendValue, **default**=180;

property TextOverlay\_AlphaBlendValue as Long

#### Description

Specifies the alpha blending value of an overlayed text selected by [TextOverlay\\_Selector](#)  
The alpha blending must be enabled by [TextOverlay\\_AlphaBlend](#)  
Look at the [Text Overlays](#) chapter for more information.

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## TextOverlay\_BkColor

### TVideoGrabber.TextOverlay\_BkColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the background color used to draw text over frames for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_BkColor: TColor **read** GetTextOverlayBkColor **write** SetTextOverlayBkColor **default** DEF\_TextOverlay\_BkColor;

\_\_property Graphics::TColor TextOverlay\_BkColor=read=GetTextOverlayBkColor, write=SetTextOverlayBkColor, **default**=16777215

Property TextOverlay\_BkColor As OLE\_COLOR

#### Description

Used to set or retrieve the background color used to draw text over frames. Useful only if [TextOverlay\\_Transparent](#) is disabled.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## TextOverlay\_Enabled

### TVideoGrabber.TextOverlay\_Enabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables drawing text over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Enabled: Boolean **read** GetTextOverlayEnabled **write** SetTextOverlayEnabled **default** DEF\_TextOverlay\_Enabled;

\_\_property **bool** TextOverlay\_Enabled=read=GetTextOverlayEnabled, write=SetTextOverlayEnabled, **default**=0

Property TextOverlay\_Enabled As Boolean

### Description

Used to enable /disable text overlay over video frames.

Note: the [frame grabber](#) must be enabled to use the set of TextOverlay properties.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## TextOverlay\_Font

### TVideoGrabber.TextOverlay\_Font

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets/retrieves the font used to draw text over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

### Declaration

**property** TextOverlay\_Font: TFont **read** GetTextOverlayFont **write** SetTextOverlayFont **default** DEF\_TextOverlay\_Font;

\_\_property Graphics::TFont \*TextOverlay\_Font=read=GetTextOverlayFont, write=SetTextOverlayFont, default=0

Property TextOverlay\_Font As Long

### Description

Used to set / retrieve the font used to draw text over video frames.

In C#.NET, use the ToHfont().ToInt32() function of the Font object. E.g.:

```
Font NewFont = new Font("Courier New", 14);
axVideoGrabberNET1.TextOverlay_Font = NewFont.ToHfont().ToInt32();
```

```
axVideoGrabberNET1.TextOverlay_Enabled = true;
axVideoGrabberNET1.StartPreview();
```

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free PDF documentation generator](#)

## TextOverlay\_FontColor

### TVideoGrabber.TextOverlay\_FontColor

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Adjusts the font color in the OCX and DLL versions for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_FontColor: TColor **read** GetTextOverlayFontColor **write** SetTextOverlayFontColor  
**default** DEF\_TextOverlay\_FontColor

**\_\_property** TColor TextOverlay\_Selector==GetTextOverlaySelector, write=SetTextOverlaySelector, **default** 0xFFFF00;

property TextOverlay\_FontColor as Long

#### Description

Used to adjust the font color in the OCX and DLL versions  
 (for Delphi and C++Builder modify directly the [TextOverlay\\_Font](#) property)

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#) [OnFrameOverlayUsingDC](#)



[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#)  
[TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#)  
[TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer

## TextOverlay\_Left

### TVideoGrabber.TextOverlay\_Left

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Left position (in pixels) where the text will be drawn over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Left: LongInt **read** GetTextOverlayLeft **write** SetTextOverlayLeft **default** DEF\_TextOverlay\_Left;

\_\_property **int** TextOverlay\_Left=read=GetTextOverlayLeft, write=SetTextOverlayLeft, **default**=0

Property TextOverlay\_Left As Long

#### Description

Used to set or retrieve the left position (in pixels) where the text will be drawn over video frames. The text will be left-padded on this position if [TextOverlay\\_Align](#) = `tf_Left`.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#)  
[TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#)  
[TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

## TextOverlay\_Right

### TVideoGrabber.TextOverlay\_Right

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Right position (in pixels) where the text will be drawn over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Right: LongInt **read** GetTextOverlayRight **write** SetTextOverlayRight **default** DEF\_TextOverlay\_Right;

\_\_property **int** TextOverlay\_Right=read=GetTextOverlayRight, write=SetTextOverlayRight, **default**=320

Property TextOverlay\_Right As Long

#### Description

Used to set or retrieve the right position (in pixels) where the text will be drawn over video frames. The text will be left-padded on this position if [TextOverlay\\_Align](#) = tf\_Right.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#)  
[TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#)  
[TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)



## TextOverlay\_Scrolling

### TVideoGrabber.TextOverlay\_Scrolling

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Enables the text scrolling for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Scrolling: Boolean **read** GetTextOverlayScrolling **write** SetTextOverlayScrolling  
**default** DEF\_TextOverlay\_Scrolling;

**\_\_property** bool TextOverlay\_Scrolling=read=GetTextOverlayScrolling, write=SetTextOverlayScrolling,  
**default**=0

property TextOverlay\_Scrolling as Boolean

#### Description

Enable the scrolling of the text overlay.

The scrolling speed is adjusted by the [TextOverlay\\_ScrollingSpeed](#) property.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKey](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

## TextOverlay\_ScrollingSpeed

### TVideoGrabber.TextOverlay\_ScrollingSpeed

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Sets the text scrolling speed for the current text overlay selected by [TextOverlay\\_Selector](#)

**Declaration**

**property** TextOverlay\_ScrollingSpeed: LongInt **read** GetTextOverlayScrollingSpeed **write** SetTextOverlayScrollingSpeed **default** DEF\_TextOverlay\_ScrollingSpeed;

**\_\_property** int TextOverlay\_ScrollingSpeed=read=GetTextOverlayScrollingSpeed, write=SetTextOverlayScrollingSpeed, **default**=0

property TextOverlay\_ScrollingSpeed as Long

**Description**

Used to adjust the scrolling speed of the text overlay.

The scrolling must be enabled first by enabling the [TextOverlay\\_Scrolling](#) property.

- TextOverlay\_ScrollingSpeed = 0 -> no scrolling
- TextOverlay\_ScrollingSpeed > 0 -> scrolling from right to left
- TextOverlay\_ScrollingSpeed < 0 -> scrolling from left to right

E.g.:

TextOverlay\_ScrollingSpeed = 1 -> slow scrolling

TextOverlay\_ScrollingSpeed = -3 -> backwards average scrolling

TextOverlay\_ScrollingSpeed = 6 -> fast scrolling

**See Also**

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

**TextOverlay\_Selector****TVideoGrabber.TextOverlay\_Selector**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects the current text overlay

**Declaration**

**property** TextOverlay\_Selector: LongInt **read** GetTextOverlaySelector **write** SetTextOverlaySelector;

**\_\_property** int TextOverlay\_Selector==GetTextOverlaySelector, write=SetTextOverlaySelector, **nodefault**};

Property TextOverlay\_Selector as Long

**Description**

Used to select a group of text overlay settings.

Up to 1000 text overlay groups of settings may be used, allowing to perform different text overlays (with different color, background, transparency, font size, etc...) at the same time.

The default text overlay settings correspond to the group 0 (TextOverlay\_Selector = 0).

*Note that TVideoGrabber does not allocate the resources for 1000 text overlays, by default it allocates only one font and one parameter set, corresponding to TextOverlay\_Selector = 0 (the resource allocation is made when TextOverlay\_Selector receives a new value never used before, see the [Text overlays](#) chapter for more explanations).*

Set TextOverlay\_Selector to the index of the text overlay properties to read or write before reading or writing their values.

It is applied to all the TextOverlay\_... properties.

E.g.:

```
procedure TfrmMainForm.Button17Click(Sender: TObject);
begin
    VideoGrabber.TextOverlay_Selector := 0;
    VideoGrabber.TextOverlay_String := 'frame number: %frame_count%';
    VideoGrabber.TextOverlay_Left := 10;
    VideoGrabber.TextOverlay_Top := 10;
    VideoGrabber.TextOverlay_Transparent := True;
    VideoGrabber.TextOverlay_Font.Color := clRed;
    VideoGrabber.TextOverlay_Font.Size := 12;
    VideoGrabber.TextOverlay_Enabled := true;

    VideoGrabber.TextOverlay_Selector := 1;
    VideoGrabber.TextOverlay_String := 'frame time: %time_100ns%';
    VideoGrabber.TextOverlay_Left := VideoGrabber.VideoWidth - 120;
    VideoGrabber.TextOverlay_Right := VideoGrabber.VideoWidth;
    VideoGrabber.TextOverlay_Top := 60;
    VideoGrabber.TextOverlay_Transparent := False;
    VideoGrabber.TextOverlay_BkColor := clBlack;
    VideoGrabber.TextOverlay_Font.Color := clWhite;
    VideoGrabber.TextOverlay_Font.Size := 8;
    VideoGrabber.TextOverlay_Enabled := true;
end;
```

in C++:

```
void __fastcall TfrmMainForm::Button2Click(TObject *Sender)

    VideoGrabber->TextOverlay_Selector = 0;
    VideoGrabber->TextOverlay_String = "frame number: %frame_count%";
    VideoGrabber->TextOverlay_Left = 10;
    VideoGrabber->TextOverlay_Top = 10;
    VideoGrabber->TextOverlay_Transparent = True;
```

```

VideoGrabber->TextOverlay_Font->Color = clRed;
VideoGrabber->TextOverlay_Font->Size = 12;
VideoGrabber->TextOverlay_Enabled = true;

VideoGrabber->TextOverlay_Selector = 1;
VideoGrabber->TextOverlay_String = "frame time: %time_100ns%";
VideoGrabber->TextOverlay_Left = VideoGrabber->VideoWidth - 120;
VideoGrabber->TextOverlay_Right = VideoGrabber->VideoWidth;
VideoGrabber->TextOverlay_Top = 60;
VideoGrabber->TextOverlay_Transparent = False;
VideoGrabber->TextOverlay_BkColor = clBlack;
VideoGrabber->TextOverlay_Font->Color = clWhite;
VideoGrabber->TextOverlay_Font->Size = 8;
VideoGrabber->TextOverlay_Enabled = true;

```

in VB:

```

Private Sub Command1_Click()
    VideoGrabberVB61.TextOverlay_Selector = 0
    VideoGrabberVB61.TextOverlay_String = "frame number: %frame_count%"
    VideoGrabberVB61.TextOverlay_Left = 10
    VideoGrabberVB61.TextOverlay_Top = 10
    VideoGrabberVB61.TextOverlay_Transparent = True
    VideoGrabberVB61.TextOverlay_FontColor = &HC0C0C0
    VideoGrabberVB61.TextOverlay_Enabled = True

    VideoGrabberVB61.TextOverlay_Selector = 1
    VideoGrabberVB61.TextOverlay_String = "frame time: %time_100ns%"
    VideoGrabberVB61.TextOverlay_Left = VideoGrabberVB61.VideoWidth - 120
    VideoGrabberVB61.TextOverlay_Right = VideoGrabberVB61.VideoWidth
    VideoGrabberVB61.TextOverlay_Top = 60
    VideoGrabberVB61.TextOverlay_Transparent = False
    VideoGrabberVB61.TextOverlay_BkColor = &H0
    VideoGrabberVB61.TextOverlay_FontColor = &HF0B0C0
    VideoGrabberVB61.TextOverlay_Enabled = True
End Sub

```

### See Also

[TCardinalDirection](#)
[TOnFrameOverlayUsingDC](#)
[TOnFrameOverlayUsingDIB](#)
[TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#)
[DrawBitmapOverFrame](#)
[GetFrameInfoString](#)
[ImageOverlay](#)
[StretchToVideoSize](#)  
[MouseWheelEventEnabled](#)
[OnFrameBitmap](#)
[OnFrameBitmapEventSynchronise](#)
[OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#)
[OnFrameOverlayUsingVIDEOHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)  
[OnMouseWheel](#)
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay](#)
[AlphaBlend](#)  
[SetImageOverlay](#)
[AlphaBlendValue](#)
[SetImageOverlay](#)
[ChromaKey](#)  
[SetImageOverlay](#)
[ChromaKeyLeewayPercent](#)
[SetImageOverlay](#)
[ChromaKeyRGBColor](#)  
[SetImageOverlay](#)
[Enabled](#)
[SetImageOverlay](#)
[Height](#)
[SetImageOverlay](#)
[LeftLocation](#)  
[SetImageOverlay](#)
[RotationAngle](#)
[SetImageOverlay](#)
[StretchToVideoSize](#)
[SetImageOverlay](#)
[TargetDisplay](#)  
[SetImageOverlay](#)
[TopLocation](#)
[SetImageOverlay](#)
[Transparent](#)
[SetImageOverlay](#)
[TransparentColorValue](#)  
[SetImageOverlay](#)
[UseTransparentColor](#)
[SetImageOverlay](#)
[Width](#)
[SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#)
[Align](#)
[SetTextOverlay](#)
[BkColor](#)
[SetTextOverlay](#)
[CustomVar](#)
[SetTextOverlay](#)
[Enabled](#)  
[SetTextOverlay](#)
[Font](#)
[SetTextOverlay](#)
[FontColor](#)
[SetTextOverlay](#)
[GradientColor](#)  
[SetTextOverlay](#)
[GradientMode](#)
[SetTextOverlay](#)
[HighResFont](#)
[SetTextOverlay](#)
[Left](#)
[SetTextOverlay](#)
[Right](#)  
[SetTextOverlay](#)
[Scrolling](#)
[SetTextOverlay](#)
[ScrollingSpeed](#)
[SetTextOverlay](#)
[Shadow](#)  
[SetTextOverlay](#)
[ShadowColor](#)
[SetTextOverlay](#)
[ShadowDirection](#)
[SetTextOverlay](#)
[String](#)  
[SetTextOverlay](#)
[TargetDisplay](#)
[SetTextOverlay](#)
[Top](#)
[SetTextOverlay](#)
[Transparent](#)
[ShapeOverlay](#)

[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## TextOverlay\_Shadow

### TVideoGrabber.TextOverlay\_Shadow

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables the shadow for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Shadow: Boolean **read** GetTextOverlayShadow **write** SetTextOverlaShadow **default** DEF\_TextOverlay\_Shadow;

\_\_property **bool** TextOverlay\_Shadow=read=GetTextOverlayShadow, write=SetTextOverlayShadow, default=0

Property TextOverlay\_Shadow As Boolean

#### Description

Enables the shadow under the text overlay.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Simplify Your Help Documentation Process with a Help Authoring Tool](#)

## TextOverlay\_ShadowColor

### TVideoGrabber.TextOverlay\_ShadowColor

[Prev](#)
[Next](#)



## **TVideoGrabber** Properties

Specifies the color of the shadow for the current text overlay selected by [TextOverlay\\_Selector](#)

### **Declaration**

**property** TextOverlay\_ShadowColor: TColor **read** GetTextOverlayShadowColor **write** SetTextOverlayShadowColor **default** DEF\_TextOverlay\_ShadowColor;

\_\_property Graphics::TColor TextOverlay\_ShadowColor=read=GetTextOverlayShadowColor, write=SetTextOverlayShadowColor, **default**=0

Property TextOverlay\_ShadowColor As OLE\_COLOR

### **Description**

Specifies the color of the shadow under the text overlay when [TextOverlay\\_Shadow](#) is enabled.

### **See Also**

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature

## **TextOverlay\_ShadowDirection**

### **TVideoGrabber.TextOverlay\_ShadowDirection**

[Prev](#)
[Next](#)

## **TVideoGrabber** Properties

Specifies the direction of the shadow for the current text overlay selected by [TextOverlay\\_Selector](#)

### **Declaration**

**property** TextOverlay\_ShadowColor: TCardinalDirection **read** GetTextOverlayShadowColor **write** SetTextOverlayShadowColor **default** DEF\_TextOverlay\_ShadowColor;

\_\_property TCardinalDirection TextOverlay\_ShadowColor=read=GetTextOverlayShadowColor, write=SetTextOverlayShadowColor, **default**=0

Property TextOverlay\_ShadowColor As TxCardinalDirection

### Description

Specifies the direction of the shadow under the text overlay when [TextOverlay\\_Shadow](#) is enabled.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Full-featured EPub generator

## TextOverlay\_String

### TVideoGrabber.TextOverlay\_String

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the text string that will be drawn over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

### Declaration

**property** TextOverlay\_String: **string** read GetTextOverlaystring **write** SetTextOverlaystring;

\_\_property wchar\_t \*TextOverlay\_String=read=GetTextOverlayString, write=SetTextOverlayString

Property TextOverlay\_String As String

### Description

Used to set or retrieve the text string that will be drawn over video frames.

The text string can be composed of several lines.

This property can be modified at any time, even if the text is currently drawn over a video frame.

Several TVideoGrabber variables can be used within the text string, delimited by a percent symbol **(2)**. When TVideoGrabber detects one of these variables the variable label is replaced by its current value:

"%sys\_time[dd/mm/yy hh:nn:ss]%" : current system date/time **(1)**

"%dv\_time[dd/mm/yy hh:nn:ss]%" : current date/time stored on the DV VCR tape **(1)**



"%time\_code%" : current DV VCR time code, if available  
 "%frame\_count%" : number of the current frame  
 "%time\_full%" : time of the current frame in hh:mm:ss:cc format  
 "%time\_sec%" : time of the current frame, in seconds with 2 decimals  
 "%time\_100ns%" : time of the current frame, in 100 nano-seconds units  
 "%custom0% to %custom9%" : up to 10 custom variables that can be set by using [SetTextOverlayCustomVar](#) .

(1) any valid [date/time format](#) is accepted between the brackets of sys\_time and dv\_time.  
 (2) the percent symbol is a reserved character. If you need to display the percent symbol itself, just duplicate it, e.g. TextOverlay\_String = "the percent symbol is %%"

Note: the [frame grabber](#) must be enabled to use TextOverlay properties.

E.g., to initialize the 2 text overlay string, that will use custom variables:

```
...
VideoGrabber.TextOverlay_Selector = 0
VideoGrabber.TextOverlay_String = "my first text uses %custom0%"
VideoGrabber.TextOverlay_Left = 10
VideoGrabber.TextOverlay_Top = 10
VideoGrabber.TextOverlay_Enabled = true

VideoGrabber.TextOverlay_Selector = 1
VideoGrabber.TextOverlay_String = "my second text uses %custom3% and %custom4%"
VideoGrabber.TextOverlay_Left = 20
VideoGrabber.TextOverlay_Top = 100
VideoGrabber.TextOverlay_Enabled = true

VideoGrabber.StartPreview
...
```

then to update the custom variables of the 1st text overlay string:

```
...
VideoGrabber.SetTextOverlayCustomVar (0, 0, "value 0")
...
```

and of the 2nd text overlay string:

```
...
VideoGrabber.SetTextOverlayCustomVar (1, 3, "value 3")
VideoGrabber.SetTextOverlayCustomVar (1, 4, "value 4")
...
```

## See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)

[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)
[SetTextOverlay](#)
[Align](#)
[SetTextOverlay](#)
[BkColor](#)
[SetTextOverlay](#)
[CustomVar](#)
[SetTextOverlay](#)
[Enabled](#)
[SetTextOverlay](#)
[Font](#)
[SetTextOverlay](#)
[FontColor](#)
[SetTextOverlay](#)
[GradientColor](#)
[SetTextOverlay](#)
[GradientMode](#)
[SetTextOverlay](#)
[HighResFont](#)
[SetTextOverlay](#)
[Left](#)
[SetTextOverlay](#)
[Right](#)
[SetTextOverlay](#)
[Scrolling](#)
[SetTextOverlay](#)
[ScrollingSpeed](#)
[SetTextOverlay](#)
[Shadow](#)
[SetTextOverlay](#)
[ShadowColor](#)
[SetTextOverlay](#)
[ShadowDirection](#)
[SetTextOverlay](#)
[String](#)
[SetTextOverlay](#)
[TargetDisplay](#)
[SetTextOverlay](#)
[Top](#)
[SetTextOverlay](#)
[Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay](#)
[Align](#)
[TextOverlay](#)
[BkColor](#)
[TextOverlay](#)
[CreateCustomFont](#)
[TextOverlay](#)
[CreateCustomFont2](#)
[TextOverlay](#)
[Enabled](#)
[TextOverlay](#)
[Font](#)
[TextOverlay](#)
[FontColor](#)
[TextOverlay](#)
[Left](#)
[TextOverlay](#)
[Right](#)
[TextOverlay](#)
[Scrolling](#)
[TextOverlay](#)
[ScrollingSpeed](#)
[TextOverlay](#)
[Selector](#)
[TextOverlay](#)
[Shadow](#)
[TextOverlay](#)
[ShadowColor](#)
[TextOverlay](#)
[ShadowDirection](#)
[TextOverlay](#)
[Top](#)
[TextOverlay](#)
[Transparent](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool

## TextOverlay\_Top

### TVideoGrabber.TextOverlay\_Top

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Top position (in pixels) where the text will be drawn over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Top: LongInt **read** GetTextOverlayTop **write** SetTextOverlayTop **default** DEF\_TextOverlay\_Top;

\_\_property **int** TextOverlay\_Top=read=GetTextOverlayTop, write=SetTextOverlayTop, **default**=0

Property TextOverlay\_Top As Long

#### Description

Used to set or retrieve the top position (in pixels) where the text will be drawn over video frames.

#### See Also

[TCardinalDirection](#)
[TOnFrameOverlayUsingDC](#)
[TOnFrameOverlayUsingDIB](#)
[TTextOverlayAlign](#)
[AdjustOverlayAspectRatio](#)
[DrawBitmapOverFrame](#)
[GetFrameInfoString](#)
[ImageOverlay](#)
[StretchToVideoSize](#)
[MouseWheelEventEnabled](#)
[OnFrameBitmap](#)
[OnFrameBitmapEventSynchronon](#)
[OnFrameOverlayUsingDC](#)
[OnFrameOverlayUsingDIB](#)
[OnFrameOverlayUsingVIDEOHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)
[OnMouseWheel](#)
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay](#)
[AlphaBlend](#)
[SetImageOverlay](#)
[AlphaBlendValue](#)
[SetImageOverlay](#)
[ChromaKey](#)
[SetImageOverlay](#)
[ChromaKeyLeewayPercent](#)
[SetImageOverlay](#)
[ChromaKeyRGBColor](#)
[SetImageOverlay](#)
[Enabled](#)
[SetImageOverlay](#)
[Height](#)
[SetImageOverlay](#)
[LeftLocation](#)
[SetImageOverlay](#)
[RotationAngle](#)
[SetImageOverlay](#)
[StretchToVideoSize](#)
[SetImageOverlay](#)
[TargetDisplay](#)
[SetImageOverlay](#)
[TopLocation](#)
[SetImageOverlay](#)
[Transparent](#)
[SetImageOverlay](#)
[TransparentColorValue](#)
[SetImageOverlay](#)
[UseTransparentColor](#)
[SetImageOverlay](#)
[Width](#)
[SetImageOverlayFromBMPFile](#)
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)
[SetTextOverlay](#)
[Align](#)
[SetTextOverlay](#)
[BkColor](#)
[SetTextOverlay](#)
[CustomVar](#)
[SetTextOverlay](#)
[Enabled](#)
[SetTextOverlay](#)
[Font](#)
[SetTextOverlay](#)
[FontColor](#)
[SetTextOverlay](#)
[GradientColor](#)
[SetTextOverlay](#)
[GradientMode](#)
[SetTextOverlay](#)
[HighResFont](#)
[SetTextOverlay](#)
[Left](#)
[SetTextOverlay](#)
[Right](#)
[SetTextOverlay](#)
[Scrolling](#)
[SetTextOverlay](#)
[ScrollingSpeed](#)
[SetTextOverlay](#)
[Shadow](#)
[SetTextOverlay](#)
[ShadowColor](#)
[SetTextOverlay](#)
[ShadowDirection](#)
[SetTextOverlay](#)
[String](#)
[SetTextOverlay](#)
[TargetDisplay](#)
[SetTextOverlay](#)
[Top](#)
[SetTextOverlay](#)
[Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay](#)
[Align](#)
[TextOverlay](#)
[BkColor](#)
[TextOverlay](#)
[CreateCustomFont](#)
[TextOverlay](#)
[CreateCustomFont2](#)
[TextOverlay](#)
[Enabled](#)
[TextOverlay](#)
[Font](#)
[TextOverlay](#)
[FontColor](#)
[TextOverlay](#)
[Left](#)
[TextOverlay](#)
[Right](#)
[TextOverlay](#)
[Scrolling](#)
[TextOverlay](#)
[ScrollingSpeed](#)
[TextOverlay](#)
[Selector](#)
[TextOverlay](#)
[Shadow](#)
[TextOverlay](#)
[ShadowColor](#)

[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## TextOverlay\_Transparent

### TVideoGrabber.TextOverlay\_Transparent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables/disables transparency of text drawn over video frames for the current text overlay selected by [TextOverlay\\_Selector](#)

#### Declaration

**property** TextOverlay\_Transparent: Boolean **read** GetTextOverlayTransparent **write** SetTextOverlayTransparent **default** DEF\_TextOverlay\_Transparent;

\_\_property **bool** TextOverlay\_Transparent=read=GetTextOverlayTransparent, write=SetTextOverlayTransparent, **default**=1

Property TextOverlay\_Top As Long

#### Description

Used to enable or disable transparency of text drawn over video frames. If enabled, [TextOverlay\\_BkColor](#) is ignored.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## TextOverlay\_VideoAlignment

### TVideoGrabber.TextOverlay\_VideoAlignment

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Relative alignment of the text overlay

#### Declaration

**property** TextOverlay\_VideoAlignment: TVideoAlignment **read** GetTextOverlayVideoAlignment **write** SetTextOverlayVideoAlignment;

**\_\_property** TVideoAlignment TextOverlay\_VideoAlignment==GetTextOverlayVideoAlignment, write=SetTextOverlayVideoAlignment, **nodefault**;

#### Description

[TVideoAlignment](#) relative alignment of the text overlay within the video frame.  
Default: oa\_LeftTop

#### See Also

[GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_Align](#)  
[SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#)  
[SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#)  
[SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## ThirdPartyDeinterlacer

### TVideoGrabber.ThirdPartyDeinterlacer

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the deinterlacer filter to use

#### Declaration

**property** ThirdPartyDeinterlacer: **string** **read** GetThirdPartyDeinterlacer **write** SetThirdPartyDeinterlacer;

**\_\_property** wchar\_t \*ThirdPartyDeinterlacer=read=GetThirdPartyDeinterlacer, write=SetThirdPartyDeinterlacer

Property ThirdPartyDeinterlacer As String

#### Description

Specifies the deinterlacer filter to be used with interlaced sources.

You can find

Set [VideoProcessing\\_Deinterlacing](#) = di\_ThirdPartyDeinterlacer to activate it.

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## TranslateMouseCoordinates

### TVideoGrabber.TranslateMouseCoordinates

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Converts mouse coordinates of [OnMouseMove](#) , [OnMouseDown](#) and [OnMouseUp](#) into their corresponding native video coordinates.

#### Declaration

**property** TranslateMouseCoordinates: Boolean **read** GetTranslateMouseCoordinates **write** SetTranslateMouseCoordinates **default** DEF\_TranslateMouseCoordinates;

\_\_property bool TranslateMouseCoordinates=read=GetTranslateMouseCoordinates, write=SetTranslateMouseCoordinates, **default**=1

Property TranslateMouseCoordinates As Boolean

### Description

This property is used when drawing over video frames using mouse coordinates returned by mouse events and the [preview zoom size](#) is not 100%.

- when this property is disabled, the mouse events return the zoomed coordinates, that do not match the native coordinates necessary to draw over frames from the [OnFrameOverlayUsingDC](#) event.
- when this property is enabled, [OnMouseMove](#) , [OnMouseDown](#) and [OnMouseUp](#) always return the native video coordinates, whatever the preview zoom size.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)

Created with the Standard Edition of HelpNDoc: Make Help Documentation a Breeze with a Help Authoring Tool

## TunerFrequency

### TVideoGrabber.TunerFrequency

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets/retrieve the TV frequency

### Declaration

**property** TunerFrequency: LongInt **read** GetTunerFrequency **write** SetTunerFrequency **default** DEF\_TunerFrequency;

\_\_property int TunerFrequency=read=GetTunerFrequency, write=SetTunerFrequency, default=-1

Property TunerFrequency as Long

## Description

Used to set / retrieve the TV frequency

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

## TunerMode

### TVideoGrabber.TunerMode

[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects the mode of the tuner.

### Declaration

**property** TunerMode: TTunerMode **read** GetTunerMode **write** SetTunerMode **default** DEF\_TunerMode;

TTunerMode TunerMode==GetTunerMode, write=SetTunerMode, **default**=0;

property TunerMode as TxTunerMode

### Description

Used to select the mode of the tuner.

The possible values are:

TV: **tm\_TV**Tuner (default)  
FM radio: **tm\_FMRadio**Tuner  
AM radio: **tm\_AMRadio**Tuner  
Digital satellite: **tm\_DigitalSatellite**Tuner

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

## TVChannel

### TVideoGrabber.TVChannel

[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves the current TV channel.

### Declaration

**property** TVChannel: LongInt **read** GetTVChannel **write** SetTVChannel;



\_\_property **int** TVChannel=read=GetTVChannel, write=SetTVChannel, **nodefault**

Property TVChannel As Long

### Description

Used to set/retrieve the current TV channel.

Be sure to select the proper country code ([TVCountryCode](#)) and tuner input type ([TVTunerInputType](#)) before.

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## TVCountryCode

### TVideoGrabber.TVCountryCode

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets / retrieves the current TV country code.

### Declaration

**property** TVCountryCode: LongInt **read** GetTVCountryCode **write** SetTVCountryCode;

\_\_property **int** TVCountryCode=read=GetTVCountryCode, write=SetTVCountryCode, **nodefault**

Property TVCountryCode As Long

### Description

Used to set / retrieve the current TV country code in which the [TV channels](#) retrieve their related TV frequencies, depending of the country.

Therefore be sure to select the proper country code before selecting a [TV channel](#).

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## TVTunerInputType

### TVideoGrabber.TVTunerInputType

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets / retrieves the current TV tuner input type.

### Declaration

**property** TVTunerInputType: TTunerInputType **read** GetTVTunerInputType **write** SetTVTunerInputType;



\_\_property TTunerInputType TVTunerInputType=read=GetTVTunerInputType, write=SetTVTunerInputType, **nodefault**

Property TVTunerInputType As TxTunerInputType

### Description

Used to set / retrieve the current TV tuner input type (antenna or cable).  
Be sure to set it before selecting a [TV channel](#).

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

## TVUseFrequencyOverrides

### TVVideoGrabber.TVUseFrequencyOverrides

[Prev](#)

[Next](#)

[TVVideoGrabber](#) [Properties](#)

Enables / disables frequency overrides.

### Declaration

**property** TVUseFrequencyOverrides: Boolean **read** GetTVUseFrequencyOverrides **write** SetTVUseFrequencyOverrides **default** DEF\_TVUseFrequencyOverrides;

\_\_property **bool** TVUseFrequencyOverrides=read=GetTVUseFrequencyOverrides, write=SetTVUseFrequencyOverrides, **default**=0

Property TVUseFrequencyOverrides As Boolean

### Description

Used to enable / disable frequency overrides that could have been set on TV channels with [TVSetChannelFrequencyOverride](#) .

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

## UniqueID

### TVVideoGrabber.UniqueID

[Prev](#)

[Next](#)

[TVVideoGrabber](#) [Properties](#)

Unique ID of the component.

## Declaration

**property** UniqueID: LongInt **read** GetUniqueID **write** SetUniqueID;

\_\_property int UniqueID=read=GetUniqueID, write=SetUniqueID, **nodefault**

Property UniqueID As Long

## Description

Unique ID of a TVideoGrabber component in the app.

Used to associate the master component to the slaves components in the master/slave [multiplexed inputs mode](#).

## See Also

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#)  
[MultiplexedRole](#) [MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation Process with HelpNDoc's Advanced Features](#)

## UseClock

### TVideoGrabber.UseClock

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Let the player graph run at the maximal speed

## Declaration

**property** UseClock: Boolean **read** GetUseClock **write** SetUseClock;

\_\_property bool UseClock=read=GetUseClock, write=SetUseClock, **nodefault**

Property UseClock As Boolean

## Description

The speed of the player graph is under control of the renderer (video renderer, audio renderer, streaming, etc).

Disabling UseClock lets the player run at maximal speed.

This is used mainly e.g. to play a clip at the processing speed, e.g. to analyse the video frames without dropping any of them because of the player speed.

Created with the Standard Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

## v360\_AspectRatio

### TVideoGrabber.v360\_AspectRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

360° video: aspect ratio of the decoded 360 video

## Declaration

**property** v360\_AspectRatio: Double **read** Getv360\_AspectRatio **write** Setv360\_AspectRatio **default** DEF\_v360\_AspectRatio;

\_\_property double v360\_AspectRatio==Getv360\_AspectRatio, write=Setv360\_AspectRatio, **default**=0;

## Description

By default the aspect ratio is calculated automatically when v360\_AspectRatio = -1.0

However it is possible to force an aspect ratio by setting a non-default value, e.g.:

- for an aspect ratio of 2:1 (2/1), set v360\_AspectRatio = 2.0
- for an aspect ratio of 16:9 (16/9), set v360\_AspectRatio = 1.7777778

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#) [v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## v360\_Enabled

### TVideoGrabber.v360\_Enabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

360° video: enables the video 360 decoding

#### Declaration

**property** v360\_Enabled: Boolean **read** Getv360\_Enabled **write** Setv360\_Enabled **default** DEF\_v360\_Enabled;

**\_\_property** bool v360\_Enabled==Getv360\_Enabled, write=Setv360\_Enabled, **default**=0;

#### Description

Enables the video 360 decoding  
Applies only to the player mode

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#) [v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's HTML5 template](#)

## v360\_MasterAngle

### TVideoGrabber.v360\_MasterAngle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

360° video: specifies the master angle

#### Declaration

**property** v360\_MasterAngle: Tv360\_Angle **read** Getv360\_MasterAngle **write** Setv360\_MasterAngle **default** DEF\_v360\_MasterAngle;

**\_\_property** Tv360\_Angle v360\_MasterAngle==Getv360\_MasterAngle, write=Setv360\_MasterAngle, **default**=1;

#### Description

Specifies the [Tv360\\_Angle](#) angle (horizontal or vertical) from which is calculated automatically the opposite angle (vertical or horizontal) when the aspect ratio is not specified (when [AspectRatioToUse](#) = -1)  
In other words, if v360\_MasterAngle = v360\_fov\_Horizontal (which is the default), v360\_fov\_Vertical is calculated automatically according to the aspect ratio of the video clip.  
Note: to control the aspect ratio yourself, set v360\_AspectRatio with a value > 0.

E.g. for an aspect ratio of 2:1, set v360\_AspectRatio = 0.5

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## v360\_MouseAction

### TVideoGrabber.v360\_MouseAction

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

360° video: specifies how the mouse can control the motion in the 360 video

#### Declaration

**property** v360\_MouseAction: TV360\_MouseAction **read** Getv360\_MouseAction **write** Setv360\_MouseAction **default** DEF\_v360\_MouseAction;

**\_\_property** TV360\_MouseAction v360\_MouseAction==Getv360\_MouseAction, write=Setv360\_MouseAction, **default**=2;

#### Description

When enabled:

- the motion of the mouse controls the point of view (horizontal / vertical)
- the mouse wheel controls the output angles (zoom)

[TV360\\_MouseAction](#) possible values:

**ma\_Disabled:** the mouse does not control the point of view.

**ma\_MouseUp:** the point of view is updated when releasing the mouse button

**ma\_MouseMove:** the point of view is updated while the mouse is moving

Note that the point of view can also be updated programmatically with [v360\\_SetYawPitchRoll](#) or [v360\\_AddYawPitchRoll](#).

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#) [v360\\_MasterAngle](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

## v360\_MouseActionPercent

### TVideoGrabber.v360\_MouseActionPercent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

360° video: percentage of displacement

#### Declaration

**property** v360\_MouseActionPercent: LongInt **read** Getv360\_MouseActionPercent **write** Setv360\_MouseActionPercent **default** DEF\_v360\_MouseActionPercent;

**\_\_property** int v360\_MouseActionPercent==Getv360\_MouseActionPercent, write=Setv360\_MouseActionPercent, **default**=10;

#### Description

Specifies the percentage of displacement of the point view, depending on the mouse movment

Default: 10%

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#) [v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Free EBook and documentation generator](#)

## VCRHorizontalLocking

### TVideoGrabber.VCRHorizontalLocking

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Sets or retrieves whether the video is from a broadcast source or from a tape source.

#### Declaration

**property** VCRHorizontalLocking: Boolean **read** GetVCRHorizontalLocking **write** SetVCRHorizontalLocking;

\_\_property **bool** VCRHorizontalLocking=read=GetVCRHorizontalLocking, write=SetVCRHorizontalLocking, **nodefault**

Property VCRHorizontalLocking As Boolean

#### Description

Used to set or retrieve whether the video is from a broadcast source or from a tape source.

**false:** the video is from a broadcast source,

**true:** the video is from a tape source.

#### See Also

[TDVCommand](#) [IsDVCommandAvailable](#) [IsHorizontalSyncLocked](#) [IsTimeCodeReaderAvailable](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## Version

### TVideoGrabber.Version

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Current version of TVideoGrabber.

#### Declaration

**property** Version: **string** **read** GetVersion **write** SetVersion **stored** false;

\_\_property wchar\_t \*Version=read=GetVersion, write=SetVersion, stored=**false**

Property Version As String

#### Description

Used to retrieve the current version of TVideoGrabber.

This property is published and visible at design time from the Object Inspector.

Created with the Standard Edition of HelpNDoc: [Free PDF documentation generator](#)

## VideoCompression\_DataRate

### TVideoGrabber.VideoCompression\_DataRate

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Specifies the data rate that should be used for video compression.

### Declaration

**property** VideoCompression\_DataRate: LongInt **read** GetVideoCompression\_DataRate **write** SetVideoCompression\_DataRate **default** DEF\_VideoCompression\_DataRate;

**\_\_property int** VideoCompression\_DataRate==GetVideoCompression\_DataRate, write=SetVideoCompression\_DataRate, **default=-1**;

### Description

Used to Specify the data rate that should be used for video compression.  
If the value specified is -1, the the default data rate of the codec will be used.

Use the [OnVideoCompressionSettings](#) event to know the default value for the current codec (see the [Software compression by using codecs](#) chapter for more explanations).

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

## VideoCompression\_KeyFrameRate

### [TVideoGrabber.VideoCompression\\_KeyFrameRate](#)

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Specifies the key-frame rate used for the [current video compressor](#).

### Declaration

**property** VideoCompression\_KeyFrameRate: LongInt **read** GetVideoCompression\_KeyFrameRate **write** SetVideoCompression\_KeyFrameRate **default** DEF\_VideoCompression\_KeyFrameRate;

**\_\_property int** VideoCompression\_KeyFrameRate=read=GetVideoCompression\_KeyFrameRate, write=SetVideoCompression\_KeyFrameRate, **default=15**

Property VideoCompression\_KeyFrameRate As Long

### Description

Specifies the desired key-frame rate for the [current video compressor](#).  
If the value specified is -1, the filter will use the default key-frame rate. If the value is zero, only the first frame will be a key frame. If the value is negative (-1), the filter will use its default key-frame rate.

Use the [OnVideoCompressionSettings](#) event to know the default value for the current codec (see the [Software compression by using codecs](#) chapter for more explanations).

### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your CHM Help Files with HelpNDoc's Advanced Customization Options](#)

## VideoCompression\_PFramesPerKeyFrame

### TVideoGrabber.VideoCompression\_PFramesPerKeyFrame

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the rate of the predicted frames per key frame for the [current video compressor](#).

#### Declaration

**property** VideoCompression\_PFramesPerKeyFrame: LongInt **read** GetVideoCompression\_PFramesPerKeyFrame **write** SetVideoCompression\_PFramesPerKeyFrame  
**default** DEF\_VideoCompression\_PFramesPerKeyFrame;

\_\_property int

VideoCompression\_PFramesPerKeyFrame=read=GetVideoCompression\_PFramesPerKeyFrame,  
write=SetVideoCompression\_PFramesPerKeyFrame, **default**=0

Property VideoCompression\_PFramesPerKeyFrame As Long

#### Description

Specifies the rate of predicted (P) frames per key frame for the [current video compressor](#).  
If the value is negative, the filter will use the default rate. If the value is negative (-1), the filter will use its default value.

Use the [OnVideoCompressionSettings](#) event to know the default value for the current codec (see the [Software compression by using codecs](#) chapter for more explanations).

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## VideoCompression\_Quality

### TVideoGrabber.VideoCompression\_Quality

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the compression quality for the [current video compressor](#).

#### Declaration

**property** VideoCompression\_Quality: Double **read** GetVideoCompression\_Quality **write** SetVideoCompression\_Quality;

\_\_property **double** VideoCompression\_Quality=read=GetVideoCompression\_Quality,  
write=SetVideoCompression\_Quality

Property VideoCompression\_Quality As Double

#### Description

Specifies compression quality for the [current video compressor](#).  
This property is expressed as a value between 0.0 and 1.0, where 1.0 indicates the best quality and 0.0



indicates the worst quality. If the value is negative (-1), the filter will use the default quality.

Use the [OnVideoCompressionSettings](#) event to know the default value for the current codec (see the [Software compression by using codecs](#) chapter for more explanations).

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## VideoCompression\_WindowSize

### TVideoGrabber.VideoCompression\_WindowSize

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the average data rate for the [current video compressor](#).

#### Declaration

**property** VideoCompression\_WindowSize: LongInt **read** GetVideoCompression\_WindowSize **write** SetVideoCompression\_WindowSize **default** DEF\_VideoCompression\_WindowSize;

**\_property** **int** VideoCompression\_WindowSize=**read**=GetVideoCompression\_WindowSize, **write**=SetVideoCompression\_WindowSize, **default**=- 1

Property VideoCompression\_WindowSize As Long

#### Description

Specifies the average data rate for the [current video compressor](#).

This property sets the number of frames over which the compressor must maintain an average data rate. For example, assuming a data rate of 100K/sec and a frame rate of 10 frames per second, if the window size is 1, then every frame will be 10K or less. If the window size is 5, then every five consecutive frames must average 10K per frame, but individual frames may exceed this size.

Use the [OnVideoCompressionSettings](#) event to know the default value for the current codec (see the [Software compression by using codecs](#) chapter for more explanations).

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

## VideoCompressor

### TVideoGrabber.VideoCompressor

[Prev](#)
[Next](#)

## [TVideoGrabber](#) [Properties](#)

Selects the current video compressor.

### Declaration

**property** VideoCompressor: LongInt **read** GetVideoCompressor **write** SetVideoCompressor;

\_\_property int VideoCompressor=read=GetVideoCompressor, write=SetVideoCompressor, **nodefault**

Property VideoCompressor As Long

### Description

Used to select the current video compressor in the [VideoCompressors](#) list.

### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Eliminate the Struggles of Documentation with a Help Authoring Tool](#)

## VideoCompressorName

### [TVideoGrabber](#).VideoCompressorName

[Prev](#)
[Next](#)

## [TVideoGrabber](#) [Properties](#)

Name of the current video compressor.

### Declaration

**property** VideoCompressorName: **string** **read** GetVideoCompressorName;

\_\_property wchar\_t \*VideoCompressorName=read=GetVideoCompressorName

Property VideoCompressorName As String

### Description

Used to retrieve the name of the current video compressor selected by [VideoCompressor](#) in the [VideoCompressors](#) list.

### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## VideoCompressors

### TVideoGrabber.VideoCompressors

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video compressors available on the current platform.

#### Declaration

function VideoCompressors: **string**;

\_\_property wchar\_t \* \_\_fastcall VideoCompressors();

property VideoCompressors as string

#### Description

Used to retrieve a string that contains the list of the video compressors available on the current platform. This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoCompressors;*

It is possible to retrieve programmatically the index of a video compressor by using the [FindIndexInListByName](#) function as follows:

*VideoGrabber.VideoCompressor := VideoGrabber.FindIndexInListByName (VideoGrabber.VideoCompressorList, VideoCompressorName);*

**Note:** when the application starts the [VideoCompressors](#) and [AudioCompressors](#) lists are populated with the codecs currently installed. To refresh the compressor lists after installing/uninstalling codecs without exiting/restarting your application, invoke [RefreshDevicesAndCompressorsLists](#).

#### See Also

[Recording methods and properties TCompressionType TOnVideoCompressionSettings AudioCompressor AudioCompressorIndex AudioCompressorName AudioCompressors AudioCompressorsCount CompressionMode CompressionType GetVideoCompressionSettings OnReencodingCompleted OnReencodingStarted OnVideoCompressionSettings RefreshDevicesAndCompressorsLists SaveCompressorSettingsToDataString SetVideoCompressionDefaults SetVideoCompressionSettings VideoCompression KeyFrameRate VideoCompression PFramesPerKeyFrame VideoCompression Quality VideoCompression WindowSize VideoCompressor VideoCompressorIndex VideoCompressorName VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)

## VideoCompressorsCount

### TVideoGrabber.VideoCompressorsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of video compressors (video codecs) available on the current platform.

#### Declaration

**property** VideoCompressorsCount: LongInt **read** GetGlobal\_VideoCompressorsCount;

\_\_property **int** VideoCompressorsCount=read=GetGlobal\_VideoCompressorsCount, **nodefault**

Function VideoCompressorsCount as Long

#### Description

Number of video compressors (video codecs) in the [VideoCompressors](#) list.

**Note:** in Delphi and C++Builder versions of the component, this property is available as a [VideoCompressorsCount](#) global variable.

**See Also**

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with a Help Authoring Tool](#)

**VideoControlSettings****TVideoGrabber.VideoControlSettings**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables or disables the [video control](#) settings.

**Declaration**

**property** VideoControlSettings: Boolean **read** GetVideoControlSettings **write** SetVideoControlSettings  
**default** DEF\_VideoControlSettings;

\_\_property **bool** VideoControlSettings=read=GetVideoControlSettings, write=SetVideoControlSettings,  
**default**=1

Property VideoControlSettings As Boolean

**Description**

Enables or disables the automatic saving and restoring of the [video control](#) settings.

**See Also**

[TVideoControl](#) [GetVideoControlMode](#) [IsVideoControlAvailable](#) [IsVideoControlModeAvailable](#)  
[SetVideoControlMode](#) [SetVideoControlMode2](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

**VideoCursor****TVideoGrabber.VideoCursor**[Prev](#)[Next](#)

[TVideoGrabber](#) [Properties](#)

Cursor displayed in the video window

**Declaration**

**property** VideoCursor: TCursor **read** GetVideoCursor **write** SetVideoCursor **default** crDefault;

\_\_property Controls::TCursor VideoCursor=read=GetVideoCursor, write=SetVideoCursor, **default**=0

Property VideoCursor As TxCursors

**Description**

Used to specify the cursor that will be displayed in the video window.

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## VideoDevice

### TVideoGrabber.VideoDevice

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Selects the current video capture device.

#### Declaration

**property** VideoDevice: LongInt **read** GetVideoDevice **write** SetVideoDevice **default** DEF\_VideoDevice;

\_\_property **int** VideoDevice=read=GetVideoDevice, write=SetVideoDevice, **default**=0

Property VideoDevice As Long

#### Description

Used to select the current video capture device. This property is an index in the [VideoDevices](#) list. When a new value is assigned to this property, the related [device-dependent values](#) are loaded from the registry and the [OnVideoDeviceSelected](#) event occurs.

It is possible to select the video capture device programmatically by its name (as it appears in the [VideoDevices](#) list) by using the [FindIndexInListByName](#) function, e.g.:

```
procedure TfrmMainForm.Button1Click(Sender: TObject);
var
    i: LongInt;
begin
    i := VideoGrabber.FindIndexInListByName (VideoGrabber.VideoDevices, 'Microsoft DV Ca
    if i > -1 then begin // if this Device exists...
        VideoGrabber.VideoDevice := i;
    end;
end;
end;
```

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

## VideoDeviceName

### TVideoGrabber.VideoDeviceName

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Name of the current video capture device.

#### Declaration

**property** VideoDeviceName: **string** **read** GetVideoDeviceName;

\_\_property **wchar\_t** \*VideoDeviceName=read=GetVideoDeviceName

Property VideoDeviceName As String

### Description

Used to retrieve the name of the video capture device selected by [VideoDevice](#) in the [VideoDevices](#) list.

### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## VideoDevices

### TVideoGrabber.VideoDevices

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video capture devices available on the current platform.

### Declaration

function VideoDevices: **string**;

\_\_property wchar\_t \* \_\_fastcall VideoDevices();

property VideoDevices as string

### Description

Used to retrieve a string that contains the list of video capture devices available on the current platform.

This list is updated if a video capture device is connected or removed (when the [OnDeviceArrivalOrRemoval](#) event occurs).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoDevices;*

*ComboBox1.ItemIndex := VideoGrabber1.VideoDevice; (\* index in the VideoDevices list \*)*

It is possible to retrieve programmatically the index of a Video capture device by using the

[FindIndexInListByName](#) function as follows:

```
VideoGrabber.VideoDevice := VideoGrabber.FindIndexInListByName (VideoGrabber.VideoDevices
```

**Note:** in Delphi and C++Builder versions of the component, this property is available as a [VideoDevices](#) global variable.

### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## VideoDevicesCount

### TVideoGrabber.VideoDevicesCount

[Prev](#)
[Next](#)

## TVideoGrabber Properties

Number of video capture devices available on the current platform.

### Declaration

**property** VideoDevicesCount: LongInt **read** GetGlobal\_VideoDevicesCount;

\_\_property **int** VideoDevicesCount=read=GetGlobal\_VideoDevicesCount, **nodefault**

Function VideoDevicesCount as Long

### Description

Number of video capture devices in the [VideoDevices](#) and [VideoDevicesId](#) lists.

**Note:** in Delphi and C++Builder versions of the component, this property is available as a [VideoDevicesCount](#) global variable.

### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## VideoDevicesId

### TVideoGrabber.VideoDevicesId

[Prev](#)

[Next](#)

#### TVideoGrabber Properties

List of the registry identifiers of the video capture devices available on the current platform.

### Declaration

function VideoDevicesId: **string**;

\_\_property wchar\_t \* \_\_fastcall VideoDevicesId();

property VideoDevicesId as string

### Description

Used to retrieve a string that contains the list of the registry identifiers of video capture devices available on the current platform.

Unlike [VideoDevices](#), **VideoDevicesId** returns the unique registry names of the video capture devices. This is useful mainly to distinguish between several video capture devices of the same brand and model on the same platform.

This list is updated if a video capture device is connected or removed (when the [OnDeviceArrivalOrRemoval](#) event occurs).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoDevicesId;*

*ComboBox1.ItemIndex := VideoGrabber1.VideoDevice; (\* index in the VideoDevicesId list \*)*

It is possible to retrieve programmatically the index of a Video capture device by using the

[FindIndexInListByName](#) function as follows:

*VideoGrabber.VideoDevice := VideoGrabber.FindIndexInListByName (VideoGrabber.VideoDevices*



**Note:** for Delphi and C++Builder versions of the component, this property is available as a [VideoDevices](#) global variable.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## VideoDoubleBuffered

### TVideoGrabber.VideoDoubleBuffered

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Activates / deactivate the double buffering in the OCX / DLL versions

#### Declaration

**property** VideoDoubleBuffered: Boolean **read** GetVideoDoubleBuffered **write** SetVideoDoubleBuffered  
**default** DEF\_DoubleBuffered;

**\_\_property** bool VideoDoubleBuffered=read=GetVideoDoubleBuffered, write=SetVideoDoubleBuffered,  
**default**=0

Property VideoDoubleBuffered as Boolean

#### Description

Used to activates / deactivate the double buffering in the OCX / DLL versions  
(in the Delphi or C++Builder versions modify the DoubleBuffered property)

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Visible](#) [DualDisplay](#) [Width](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize your documentation process with HelpNDoc's online capabilities](#)

## VideoFormat

### TVideoGrabber.VideoFormat

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Properties](#)

Index of the video format in the VideoFormats list

#### Declaration

**property** VideoFormat: LongInt **read** GetVideoFormat **write** SetVideoFormat;

\_\_**property** int VideoFormat==GetVideoFormat, write=SetVideoFormat, **nodefault**};

Property VideoFormat as Long

#### Description

Used to select a video format in the [VideoFormats](#) list

See [Video Sizes](#)

#### Important:

The current video size index is not the same from a video capture device to the other. It is reloaded when the video capture device is selected by assigning the [VideoDevice](#) property.

- any control displaying this value should re-read it from the [OnVideoDeviceSelected](#) event (that occurs when the video device is selected)
- be sure that the video capture device has been selected (by assigning the [VideoDevice](#) property) before setting this property

See [device-dependent properties](#)

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

---

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

---

## VideoFormats

### TVideoGrabber.VideoFormats

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video formats available on the current video capture device

#### Declaration

**property** VideoFormats: string **read** GetVideoFormats;

\_\_**property** wchar\_t \*VideoFormats==GetVideoFormats;

Property VideoFormats as String

#### Description

Used to retrieve a string that contains the list of the video formats available on the current video capture device.

This list is updated when the [OnVideoDeviceSelected](#) event occurs (when a video capture device is selected with [VideoDevice](#) ).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoFormats;*

*ComboBox1.ItemIndex := VideoGrabber1.VideoFormat;*

Note: as the video formats list is not the same from a video capture device to the other, any code using this value should re-read it from the [OnVideoDeviceSelected](#) event (when a [video capture device](#) is selected).

See the [Video sizes](#) chapter for more information.

## VideoFormatsCount

### TVideoGrabber.VideoFormatsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of video formats

#### Declaration

**property** VideoFormatsCount: LongInt **read** GetVideoFormatsCount;

\_\_**property** int VideoFormatsCount==GetVideoFormatsCount, **nodefault**};

property VideoFormatsCount as Long

#### Description

Number of video formats available in the [VideoFormats](#) list for the current video capture device.

## VideoFromImages\_BitmapsSortedBy

### TVideoGrabber.VideoFromImages\_BitmapsSortedBy

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies how the bitmap or JPEG files are sorted.

#### Declaration

**property** VideoFromImages\_BitmapsSortedBy: TFileSort **read** GetVideoFromImages\_BitmapsSortedBy **write** SetVideoFromImages\_BitmapsSortedBy **default** DEF\_VideoFromImages\_BitmapsSortedBy;

\_\_**property** TFileSort VideoFromImages\_BitmapsSortedBy=read=GetVideoFromImages\_BitmapsSortedBy, write=SetVideoFromImages\_BitmapsSortedBy, **default**=0

Property VideoFromImages\_BitmapsSortedBy As TFileSort

#### Description

Used to specify in which order the BMP or JPEG files located in [VideoFromImages\\_SourceDirectory](#) will be recorded.

The value is a [TFileSort](#) type.

#### Note: to sort correctly numeric sequences of bitmaps:

- use fs\_NameAsc or fs\_NameDesc,
- be sure to make the name of the images"zero padded". E.g.:

```
001.bmp
002.bmp
003.bmp
...
009.bmp
010.bmp
011.bmp
...
098.bmp
099.bmp
```

100.bmp  
101.bmp  
...

#### See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps](#) [SendImageToVideoFromBitmaps2](#)  
[VideoFromImages](#) [RepeatIndefinitely](#) [VideoFromImages](#) [SourceDirectory](#)  
[VideoFromImages](#) [TemporaryFile](#)

Created with the Standard Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

## VideoFromImages\_RepeatIndefinitely

**TVideoGrabber.VideoFromImages\_RepeatIndefinitely** [Prev](#) [Next](#)  
[TVideoGrabber](#) [Properties](#)

Specifies if TVideoGrabber must restart from the 1st image when the last image is reached.

#### Declaration

**property** VideoFromImages\_RepeatIndefinitely: Boolean **read** GetVideoFromImages\_RepeatIndefinitely **write** SetVideoFromImages\_RepeatIndefinitely **default** DEF\_VideoFromImages\_RepeatIndefinitely;

\_\_property **bool** VideoFromImages\_RepeatIndefinitely=read=GetVideoFromImages\_RepeatIndefinitely, write=SetVideoFromImages\_RepeatIndefinitely, **default**=0

Property VideoFromImages\_RepeatIndefinitely As Boolean

#### Description

Used to specify if TVideoGrabber must restart from the 1st image when the last image is reached in the [VideoFromImages\\_SourceDirectory](#) directory

#### See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps](#) [SendImageToVideoFromBitmaps2](#)  
[VideoFromImages](#) [BitmapsSortedBy](#) [VideoFromImages](#) [SourceDirectory](#)  
[VideoFromImages](#) [TemporaryFile](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## VideoFromImages\_SourceDirectory

**TVideoGrabber.VideoFromImages\_SourceDirectory** [Prev](#) [Next](#)  
[TVideoGrabber](#) [Properties](#)

Specifies the directory where the image used to create the video clip are located.

#### Declaration

**property** VideoFromImages\_SourceDirectory: **String** **read** GetVideoFromImages\_SourceDirectory **write** SetVideoFromImages\_SourceDirectory **stored** False;

\_\_property wchar\_t \*VideoFromImages\_SourceDirectory=read=GetVideoFromImages\_SourceDirectory, write=SetVideoFromImages\_SourceDirectory, stored=**false**

Property VideoFromImages\_SourceDirectory As String

#### Description

Used to specify the directory in which the image used to create the video clip are located.

#### See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps](#) [SendImageToVideoFromBitmaps2](#) [VideoFromImages](#) [BitmapsSortedBy](#) [VideoFromImages](#) [RepeatIndefinitely](#) [VideoFromImages](#) [TemporaryFile](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation Process with HelpNDoc's Advanced Features](#)

## VideoFromImages\_TemporaryFile

### TVideoGrabber.VideoFromImages\_TemporaryFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the full qualified path and file name of the temporary file.

#### Declaration

**property** VideoFromImages\_TemporaryFile: **String** **read** GetVideoFromImages\_TemporaryFile **write** SetVideoFromImages\_TemporaryFile;

\_\_property wchar\_t \*VideoFromImages\_TemporaryFile=read=GetVideoFromImages\_TemporaryFile, write=SetVideoFromImages\_TemporaryFile

Property VideoFromImages\_TemporaryFile As String

#### Description

Specifies the full qualified path and file name of the temporary file in which the BMP or JPEG files will be encoded before creating the video clip.

#### See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps](#) [SendImageToVideoFromBitmaps2](#) [VideoFromImages](#) [BitmapsSortedBy](#) [VideoFromImages](#) [RepeatIndefinitely](#) [VideoFromImages](#) [SourceDirectory](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

## VideoHeight

### TVideoGrabber.VideoHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Retrieves the height of the native video source, in pixels.

#### Declaration

**property** VideoHeight: LongInt **read** GetVideoHeight;

\_\_property int VideoHeight=read=GetVideoHeight, **nodefault**

Property VideoHeight As Long

#### Description

Used to retrieve the height of the native video source, in pixels.

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

## VideoHeight\_PreferedAspectRatio

### TVideoGrabber.VideoHeight\_PreferedAspectRatio

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Retrieves the preferred display height of the video clip, in pixels.

#### Declaration

**property** VideoHeight\_PreferedAspectRatio: LongInt **read** GetVideoHeight\_PreferedAspectRatio;

**\_\_property** int VideoHeight\_PreferedAspectRatio=read=GetVideoHeight\_PreferedAspectRatio,  
**nodefault**

Property VideoHeight\_PreferedAspectRatio As Long

#### Description

Used to retrieve the preferred display height of the video clip, in pixels.

It can be different of the [VideoHeight](#) property if the video clip has a display aspect ratio (DAR) different of the source aspect ratio (SAR)

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TVideoRenderer](#) [TOnPlayerBufferingData](#) [AdjustPixelAspectRatio](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [IsVideoPortAvailable](#) [Last\\_Clip](#) [Played](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [MP4NeedsReindexing](#) [OnLeavingFullScreen](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SetParentWindow](#) [SetWindowTransparency](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoDoubleBuffered](#) [VideoPlayableWhileRecording](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

## VideoInput

### TVideoGrabber.VideoInput

[Prev](#)[Next](#)[TVideoGrabber](#) [Properties](#)

Selects the video input.

#### Declaration

**property** VideoInput: LongInt **read** GetVideoInput **write** SetVideoInput;

\_\_property **int** VideoInput=read=GetVideoInput, write=SetVideoInput, **nodefault**

Property VideoInput As Long

#### Description

Selects the video input.

This property is an index in the [VideoInputs](#) list, in the 0 ... [VideoInputsCount](#) - 1 range.

Important:

The current video size index is not the same from a video capture device to the other. It is reloaded when the video capture device is selected by assigning the [VideoDevice](#) property.

- any control displaying this value should re-read it from the [OnVideoDeviceSelected](#) event (that occurs when the video device is selected)
- be sure that the video capture device has been selected (by assigning the [VideoDevice](#) property) before setting this property

See [VideoInputs](#) and [device-dependent properties](#)

#### See Also

[Video inputs](#) [VideoInputIndex](#) [VideoInputs](#) [VideoInputsCount](#)

---

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

## VideoInputs

### TVideoGrabber.VideoInputs

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video inputs available on the current video capture device

#### Declaration

**property** VideoInputs: **string** **read** GetVideoInputs;

\_\_property **wchar\_t** \*VideoInputs=read=GetVideoInputs

Property VideoInputs As String

#### Description

Used to retrieve a string that contains the list of the video inputs available on the current video capture device.

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoInputs;*

*ComboBox1.ItemIndex := VideoGrabber1.VideoInput;*

Note: as this list is not the same from a video capture device to the other, any code using this value should re-read it from the [OnVideoDeviceSelected](#) event (when a [video capture device](#) is selected).

#### See Also

[Video inputs](#) [VideoInput](#) [VideoInputIndex](#) [VideoInputsCount](#)

---

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)

---



## VideoInputsCount

### TVideoGrabber.VideoInputsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of video inputs.

#### Declaration

**property** VideoInputsCount: LongInt **read** GetVideoInputsCount;

\_\_**property** **int** VideoInputsCount=read=GetVideoInputsCount, **nodefault**

Property VideoInputsCount As Long

#### Description

Number of video inputs in the [VideoInputs](#) list for the current video capture device.

#### See Also

[Video inputs](#) [VideoInput](#) [VideoInputIndex](#) [VideoInputs](#)

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

## VideoPlayableWhileRecording

### TVideoGrabber.VideoPlayableWhileRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Allow playing a video clip being recorded

#### Declaration

**property** VideoPlayableWhileRecording: Boolean **read** GetVideoPlayableWhileRecording **write** SetVideoPlayableWhileRecording **default** DEF\_VideoPlayableWhileRecording;

\_\_**property** **bool** VideoPlayableWhileRecording==GetVideoPlayableWhileRecording,  
write=SetVideoPlayableWhileRecording, **default**=0;

#### Description

When enabled, a .MP4 or .MKV video currently being recorded can be opened in the player of another TVideoGrabber instance or in a third-party player.

#### Note:

- for IP cameras and URL sources being recorded in native format, this requires the Datastead RTSP/RTMP/HTTP/ONVIF Source Filter to be installed.
- for other sources the Datastead Multipurpose Encoder must be installed

#### See Also

[Player features](#) [Recording methods and properties](#) [TAVIMuxConfig](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnPlayerBufferingData](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioChannelRenderMode](#) [AudioRecording](#) [AudioStreamNumber](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [Encoder](#) [SetInt](#) [FastForwardPlayer](#) [HoldRecording](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [IsRecordingPaused](#) [Last\\_Clip\\_Played](#) [Last\\_Recording\\_FileName](#) [MP4NeedsReindexing](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#)

[OpenPlayerAtTimePositions](#) [PausePlayer](#) [PauseRecording](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [ShowDialog](#) [SourceStream](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopPlayer](#) [StopRecording](#) [StoragePath](#) [SynchronizationRole](#) [Synchronized](#) [SyncPreview](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

## VideoProcessing\_Brightness

### TVideoGrabber.VideoProcessing\_Brightness

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Modifies the brightness of the video stream on which the [frame grabber](#) is applied.

#### Declaration

**property** VideoProcessing\_Pixellization: LongInt **read** GetVideoProcessingPixellization **write** SetVideoProcessingPixellization **default** DEF\_VideoProcessing\_Pixellization;

\_\_**property int** VideoProcessing\_Pixellization=read=GetVideoProcessingPixellization, write=SetVideoProcessingPixellization, **default**=0

Property VideoProcessing\_Pixellization As Long

#### Description

Used to modify the brightness value of the video stream on which the [frame grabber](#) is applied. In the -128..+128 range.

#### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## VideoProcessing\_Contrast

### TVideoGrabber.VideoProcessing\_Contrast

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Modifies the contrast value of the video stream on which the [frame grabber](#) is applied.

#### Declaration

**property** VideoProcessing\_Contrast: LongInt **read** GetVideoProcessingContrast **write** SetVideoProcessingContrast **default** DEF\_VideoProcessing\_Contrast;

\_\_**property int** VideoProcessing\_Contrast=read=GetVideoProcessingContrast,

write=SetVideoProcessingContrast, **default**=0

Property VideoProcessing\_Contrast As Long

### Description

Used to modify the contrast value of the video stream on which the [frame grabber](#) is applied. In the -128..+128 range.

### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

## VideoProcessing\_Deinterlacing

### TVideoGrabber.VideoProcessing\_Deinterlacing

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enable the deinterlacing of the video stream.

### Declaration

**property** VideoProcessing\_Deinterlacing: TVideoDeinterlacing **read** GetVideoProcessingDeinterlacing **write** SetVideoProcessingDeinterlacing **default** DEF\_VideoProcessing\_Deinterlacing;

\_\_property TVideoDeinterlacing VideoProcessing\_Deinterlacing=read=GetVideoProcessingDeinterlacing, write=SetVideoProcessingDeinterlacing, **default**=0

Property VideoProcessing\_Deinterlacing As TxVideoDeinterlacing

### Description

This [TVideoDeinterlacing](#) property is used to enable/disable the deinterlacing of the video stream.

See the [Deinterlacing](#) chapter.

### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

## VideoProcessing\_FlipHorizontal

### TVideoGrabber.VideoProcessing\_FlipHorizontal

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Flips horizontally the video stream on which the [frame grabber](#) is applied.

### Declaration

**property** VideoProcessing\_FlipHorizontal: Boolean **read** GetVideoProcessingLeftRight **write** SetVideoProcessingLeftRight **default** DEF\_VideoProcessing\_FlipHorizontal;

\_\_property **bool** VideoProcessing\_FlipHorizontal=read=GetVideoProcessingLeftRight, write=SetVideoProcessingLeftRight, **default**=0

Property VideoProcessing\_FlipHorizontal As Boolean

#### Description

When enabled, video frames are flipped horizontally on the video stream on which the [frame grabber](#) is applied.

#### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## VideoProcessing\_FlipVertical

### TVideoGrabber.VideoProcessing\_FlipVertical

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Flips vertically the video stream on which the [frame grabber](#) is applied.

#### Declaration

**property** VideoProcessing\_FlipVertical: Boolean **read** GetVideoProcessingTopDown **write** SetVideoProcessingTopDown **default** DEF\_VideoProcessing\_FlipVertical;

\_\_property **bool** VideoProcessing\_FlipVertical=read=GetVideoProcessingTopDown, write=SetVideoProcessingTopDown, **default**=0

Property VideoProcessing\_FlipVertical As Boolean

#### Description

When enabled, video frames are flipped vertically on the video stream on which the [frame grabber](#) is applied.

#### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Free EBook and documentation generator](#)

## VideoProcessing\_GrayScale

### TVideoGrabber.VideoProcessing\_GrayScale

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Changes colors in greyscale on the video stream on which the [frame grabber](#) is applied.

#### Declaration

**property** VideoProcessing\_GrayScale: Boolean **read** GetVideoProcessingGrayScale **write** SetVideoProcessingGrayScale **default** DEF\_VideoProcessing\_GrayScale;

```
__property bool VideoProcessing_GrayScale=read=GetVideoProcessingGrayScale,
write=SetVideoProcessingGrayScale, default=0
```

Property VideoProcessing\_GrayScale As Boolean

#### Description

When enabled, colors are changed in grayscale on the video stream on which the [frame grabber](#) is applied.

#### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

---

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

---

## VideoProcessing\_Hue

### TVideoGrabber.VideoProcessing\_Hue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Modifies the hue of the video stream on which the [frame grabber](#) is applied.

#### Declaration

```
property VideoProcessing_Hue: LongInt read GetVideoProcessingHue write SetVideoProcessingHue
default DEF_VideoProcessing_Hue;
```

```
__property int VideoProcessing_Hue=read=GetVideoProcessingHue, write=SetVideoProcessingHue,
default=0
```

Property VideoProcessing\_Hue As Long

#### Description

Used to modify the hue value of the video stream on which the [frame grabber](#) is applied.

#### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

---

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

---

## VideoProcessing\_InvertColors

### TVideoGrabber.VideoProcessing\_InvertColors

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Inverts the colors of the video stream on which the [frame grabber](#) is applied.

#### Declaration

```
property VideoProcessing_InvertColors: Boolean read GetVideoProcessingInvertColors write
SetVideoProcessingInvertColors default DEF_VideoProcessing_InvertColors;
```

\_\_property **bool** VideoProcessing\_InvertColors=read=GetVideoProcessingInvertColors, write=SetVideoProcessingInvertColors, **default**=0

Property VideoProcessing\_InvertColors As Boolean

### Description

When enabled, colors are inverted on the video stream on which the [frame grabber](#) is applied.

### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

## VideoProcessing\_Pixellization

### TVideoGrabber.VideoProcessing\_Pixellization

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to pixellize the video frames

### Declaration

**property** VideoProcessing\_Pixellization: LongInt **read** GetVideoProcessingPixellization **write** SetVideoProcessingPixellization **default** DEF\_VideoProcessing\_Pixellization;

\_\_property **int** VideoProcessing\_Pixellization==GetVideoProcessingPixellization, write=SetVideoProcessingPixellization, **default**=1

Property VideoProcessing\_Pixellization as Long

### Description

This property is used to create a pixellized effect.

Assign to this property specify a pixel size > 1.

The [FrameGrabber](#) must be enabled.

Default value: 1

### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

## VideoProcessing\_Rotation

### TVideoGrabber.VideoProcessing\_Rotation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)



Specifies if the video image must be rotated and/or mirrored, by 90, 180 or 270 degrees.

#### Declaration

**property** VideoProcessing\_Rotation: TVideoRotation **read** GetVideoProcessingRotation **write** SetVideoProcessingRotation **default** DEF\_VideoProcessing\_Rotation;

\_\_property TVideoRotation VideoProcessing\_Rotation=read=GetVideoProcessingRotation, write=SetVideoProcessingRotation, **default**=0

Property VideoProcessing\_Rotation As TVideoRotation

#### Description

This [TVideoRotation](#) property is used to specify if the video image must be rotated and/or mirrored, by 90, 180 or 270 degrees.

**rt\_0\_deg** : no rotation, not mirrored

**rt\_90\_deg** : 90° rotation

**rt\_180\_deg** : 180° rotation

**rt\_270\_deg** : 270° rotation

**rt\_0\_deg\_mirror** : no rotation, mirrored

**rt\_90\_deg\_mirror** : 90° rotation, mirrored

**rt\_180\_deg\_mirror** : 180° rotation, mirrored

**rt\_270\_deg\_mirror** : 270° rotation, mirrored

**rt\_custom\_angle** : rotation of a [VideoProcessing\\_RotationCustomAngle](#) value

**rt\_custom\_angle\_mirror\_vert** : rotation of a [VideoProcessing\\_RotationCustomAngle](#) value, with vertical mirroring

**rt\_custom\_angle\_mirror\_horz** : rotation of a [VideoProcessing\\_RotationCustomAngle](#) value, with horizontal mirroring

**rt\_custom\_angle\_mirror\_both** : rotation of a [VideoProcessing\\_RotationCustomAngle](#) value, with vertical and horizontal mirroring

*Remark: the rt\_custom\_angle... settings require a lot of CPU, if you get dropped frames use a lower video size.*

#### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## VideoProcessing\_RotationCustomAngle

### TVideoGrabber.VideoProcessing\_RotationCustomAngle

[Prev](#)

[Next](#)

e

[TVideoGrabber](#) [Properties](#)

Specifies the angle of rotation of the video image.

#### Declaration

**property** VideoProcessing\_Rotation: TVideoRotation **read** GetVideoProcessingRotation **write** SetVideoProcessingRotation **default** DEF\_VideoProcessing\_Rotation;

\_\_property double VideoProcessing\_RotationCustomAngle==GetVideoProcessingRotationCustomAngle, write=SetVideoProcessingRotationCustomAngle;

property VideoProcessing\_RotationCustomAngle as Double



## Description

Used to specify the angle of rotation of the video image, in degrees, when using the **rt\_custom\_Angle...** settings of the [VideoProcessing\\_Rotation](#) property.

## See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## VideoProcessing\_Saturation

### TVideoGrabber.VideoProcessing\_Saturation

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Modifies the saturation of the video stream on which the [frame grabber](#) is applied.

## Declaration

**property** VideoProcessing\_Saturation: LongInt **read** GetVideoProcessingSaturation **write** SetVideoProcessingSaturation **default** DEF\_VideoProcessing\_Saturation;

\_\_property **int** VideoProcessing\_Saturation=read=GetVideoProcessingSaturation, write=SetVideoProcessingSaturation, **default**=0

Property VideoProcessing\_Saturation As Long

## Description

Used to modify the saturation value of the video stream on which the [frame grabber](#) is applied. In the -64..+64 range.

## See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer](#)

## VideoQualitySettings

### TVideoGrabber.VideoQualitySettings

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Enables or disables the [video quality](#) settings.

## Declaration

**property** VideoQualitySettings: Boolean **read** GetVideoQualitySettings **write** SetVideoQualitySettings **default** DEF\_VideoQualitySettings;

\_\_property **bool** VideoQualitySettings=read=GetVideoQualitySettings, write=SetVideoQualitySettings, **default**=1

Property VideoQualitySettings As Boolean

## Description

Used to enable or disable the automatic saving / restoring of [video quality](#) settings.

## See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#) [VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## VideoRenderer

### TVideoGrabber.VideoRenderer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the video renderer used for the video window.

## Declaration

**property** VideoRenderer: TVideoRenderer **read** GetVideoRenderer **write** SetVideoRenderer **default** DEF\_VideoRenderer;

\_\_property TVideoRenderer VideoRenderer=read=GetVideoRenderer, write=SetVideoRenderer, **default**=0

Property VideoRenderer As TxVideoRenderer

## Description

Used to specify the [TVideoRenderer](#) renderer that performs rendering in the video window.

## See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## VideoRendererExternal

### TVideoGrabber.VideoRendererExternal

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enable third-party video renderers (usually to a second monitor)

## Declaration

**property** VideoRendererExternal: TVideoRendererExternal **read** GetVideoRendererExternal **write** SetVideoRendererExternal **default** DEF\_VideoRendererExternal;

\_\_property TVideoRendererExternal VideoRendererExternal=read=GetVideoRendererExternal, write=SetVideoRendererExternal, **default**=0

Property VideoRendererExternal As TxVideoRendererExternal

## Description

Used to enable third-party video renderers (usually rendered to a second monitor).

The current allowed values are:

**vre\_None:**

no external video renderer (default value)

**vre\_BlackMagic\_Decklink:**

external renderer on the [BlackMagic Decklink](#) card.

**vre\_Matrox\_PRO:**

external renderer on [Matrox](#) cards.

**vre\_Decklink\_SD, vre\_Decklink\_Extreme:**

*deprecated.*

**vre\_Pinnacle\_MovieBoard:**

external renderer on the [Pinnacle MovieBoard](#) card

*Feel free to contact our support at [support@datastead.com](mailto:support@datastead.com) if your video card has an external renderer that is not supported by the current version*

See also [VideoRendererExternalIndex](#)

**See Also**

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

---

Created with the Standard Edition of HelpNDoc: Transform Your Word Document into a Professional eBook with HelpNDoc

---

## VideoRendererExternalIndex

### TVideoGrabber.VideoRendererExternalIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Index of the external video renderer to use

**Declaration**

**property** VideoRendererExternalIndex: LongInt **read** GetVideoRendererExternalIndex **write** SetVideoRendererExternalIndex;

**\_\_property** int VideoRendererExternalIndex=read=GetVideoRendererExternalIndex, write=SetVideoRendererExternalIndex, **default**

Property VideoRendererExternalIndex As Long

**Description**

Index of the external video renderer to use.

Default value: -1 (the first card will be used)

This feature is useful when more than one external video renderer are installed, to let you specify what renderer should be used.

E.g. let's suppose a PC where 4 Blackmagic Decklink cards are installed:

**e.g. to select the 1st Decklink card:**

VideoGrabber.[VideoRendererExternal](#) = vre\_BlackMagic\_Decklink  
VideoGrabber.VideoRendererExternalIndex = 0

**e.g. to select the 2nd Decklink card:**

VideoGrabber.[VideoRendererExternal](#) = vre\_BlackMagic\_Decklink  
VideoGrabber.VideoRendererExternalIndex = 1

etc...

**See Also**

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow SetWindowTransparency VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Experience the Power and Simplicity of HelpNDoc's User Interface](#)

## VideoRendererPriority

### TVideoGrabber.VideoRendererPriority

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specify whether the quality or speed must be favored.

**Declaration**

**property** VideoRendererPriority: TVideoRendererPriority **read** GetVideoRendererPriority **write** SetVideoRendererPriority **default** DEF\_VideoRendererPriority;

**\_\_property** TVideoRendererPriority VideoRendererPriority==GetVideoRendererPriority, write=SetVideoRendererPriority, **default**=2;

**Description**

Possible values here: [TVideoRendererPriority](#)

**See Also**

[TVideoRenderer TOnFrameProgress TOnResizeVideo AdjustPixelAspectRatio AutoRefreshPreview BackgroundColor Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width](#)

[DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#)  
[DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#)  
[DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#)  
[DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#)  
[DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#)  
[DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#)  
[ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [IsVideoQualityAvailable](#)  
[IsVideoQualitySettingAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnFrameProgress](#)  
[OnKeyPress](#) [OnLeavingFullScreen](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#)  
[ResumePreview](#) [SetParentWindow](#) [SetVideoQuality](#) [SetWindowTransparency](#) [StartPreview](#)  
[StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoDoubleBuffered](#) [VideoHeight](#)  
[VideoHeight\\_PreferedAspectRatio](#) [VideoQualityAuto](#) [VideoQualityDefault](#) [VideoQualityMax](#)  
[VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRenderer](#) [VideoRendererExternal](#)  
[VideoRendererExternalIndex](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [VideoWidth\\_PreferedAspectRatio](#)  
[Visible](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

## VideoSize

### TVideoGrabber.VideoSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Selects a video size.

#### Declaration

**property** VideoSize: LongInt **read** GetVideoSize **write** SetVideoSize;

\_\_\_property **int** VideoSize=read=GetVideoSize, write=SetVideoSize, **nodefault**

Property VideoSize As Long

#### Description

Used to select a video size in the [VideoSizes](#) list.  
See [VideoSizes](#) .

#### Important:

The current video size index is not the same from a video capture device to the other. It is reloaded when the video capture device is selected by assigning the [VideoDevice](#) property.

- any control displaying this value should re-read it from the [OnVideoDeviceSelected](#) event (that occurs when the video device is selected)
- be sure that the video capture device has been selected (by assigning the [VideoDevice](#) property) before setting this property

See [device-dependent properties](#)

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#)  
[GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#)  
[VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## VideoSizes

### TVideoGrabber.VideoSizes

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

**ber**

List of the video sizes available on the current video capture device

**Declaration**

**property** VideoSizes: **string read** GetVideoSizes;

\_\_property wchar\_t \*VideoSizes=read=GetVideoSizes

Property VideoSizes As String

**Description**

Used to retrieve a string that contains the list of the video sizes available on the current video capture device.

This list is updated when the [OnVideoDeviceSelected](#) event occurs (when a video capture device is selected with [VideoDevice](#) ).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoSizes;*

*ComboBox1.ItemIndex := VideoGrabber1.VideoSize;*

Note: as the video sizes list is not the same from a video capture device to the other, any code using this value should re-read it from the [OnVideoDeviceSelected](#) event (when a [video capture device](#) is selected). See the [Video sizes](#) chapter for more information.

**See Also**

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

**VideoSizesCount****TVideoGrabber.VideoSizesCount**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of video sizes.

**Declaration**

**property** VideoSizesCount: LongInt **read** GetVideoSizesCount;

\_\_property **int** VideoSizesCount=read=GetVideoSizesCount, **nodefault**

Property VideoSizesCount As Long

**Description**

Number of video sizes in the [VideoSizes](#) list for the current video capture device.

**See Also**

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with a Help Authoring Tool](#)

**VideoSource****TVideoGrabber.VideoSource**
[Prev](#)
[Next](#)

## **TVideoGrabber** **Properties**

Specifies the video source for [preview](#) and [recording](#).

### **Declaration**

**property** VideoSource: TVideoSource **read** GetVideoSource **write** SetVideoSource **default** DEF\_VideoSource;

\_\_property TVideoSource VideoSource=read=GetVideoSource, write=SetVideoSource, **default**=0

Property VideoSource As TxVideoSource

### **Description**

This [TVideoSource](#) property is used to specify the video source used for [preview](#) and [recording](#):

#### **vs\_VideoCaptureDevice:**

the video source is the [current video capture device](#).

#### **vs\_ScreenRecording:**

the video source is the screen. See the [Screen recording](#) chapter.

#### **vs\_VideoFileOrURL:**

the video source is a video clip specified by the [VideoSource\\_FileOrURL](#) property.

#### **vs\_JPEGsOrBitmaps:**

the video source are bitmap handles, or BMP files or JPEG files of the same format passed to the [OnVideoFromBitmaps\\_NextFrameNeeded](#) event.

See [Video clips built on the fly by passing bitmap handles, BMP or JPEG files](#).

#### **vs\_IPCamera:**

the video source is an IP camera or a IP video server. See the [IP Cameras - IP Webcams - IP video servers](#) chapter.

#### **vs\_Mixer:**

configures the TVideoGrabber component as a [video mixer](#), the video source(s) is(are) other(s) TVideoGrabber component(s)

#### **vs\_VideoFromImages:**

the video source is a set of bitmaps (BMP or JPEG files). See [Video clip from bitmaps: Overview](#).

#### **vs\_ThirdPartyFilter:**

a third-party DirectShow filter is used as video source

#### **vs\_StreamInterface:**

the source is raw stream data (e.g. h264) pushed through the StreamInterface\_PushData function

### **See Also**

[Video sources supported for preview and recording](#) [VideoSources](#) [VideoSourcesCount](#) [TVideoSource](#)

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## **VideoSource\_FileOrURL**

### **TVideoGrabber.VideoSource\_FileOrURL**

[Prev](#)

[Next](#)

## **TVideoGrabber** **Properties**

Specifies the source video clip to use when [VideoSource](#) = vs\_VideoFileOrURL

### **Declaration**

**property** VideoSource\_FileOrURL: **String** **read** GetVideoSource\_FileOrURL **write** SetVideoSource\_FileOrURL;



```
__property wchar_t *VideoSource_FileOrURL=read=GetVideoSource_FileOrURL,
write=SetVideoSource_FileOrURL
```

Property VideoSource\_FileOrURL As String

### Description

Specifies the source video clip to use as live source (instead of a video capture device or a web cam) when [VideoSource](#) = vs\_VideoFileOrURL

Can be:

- a file located on a local drive, a network drive
- or a streaming video clip specified by its URL
- a playlist.

*E.g. to record a video file:*

```
videograbber.VideoSource = vs_VideoFileOrURL
videograbber.VideoSource_FileOrURL = 'myclip.avi'
videograbber.StartRecording()
```

*E.g. to record a playlist:*

```
videograbber.Playlist (pl_Clear, '')
videograbber.Playlist (pl_Add, 'vg000004.avi')
videograbber.Playlist (pl_Add, 'vg000002.avi')
videograbber.Playlist (pl_Add, ...
videograbber.Playlist (pl_Add, ...
videograbber.VideoSource = vs_VideoFileOrURL
videograbber.VideoSource_FileOrURL = 'PLAYLIST'
videograbber.StartRecording()
```

- alternatively you can invoke [StartPreview](#) instead of [StartRecording](#) to just play the clip (without trackbar handling, if you need the trackbar handling use the [Player](#) features)

- look at the [Recording methods and properties](#) chapter for information about the recording settings

### See Also

[VideoSource\\_FileOrURL\\_StartTime](#) [VideoSource\\_FileOrURL\\_StopTime](#)

---

Created with the Standard Edition of HelpNDoc: [Maximize Your Reach: Convert Your Word Document to an ePub or Kindle eBook](#)

---

## VideoSource\_FileOrURL\_StartTime

### TVideoGrabber.VideoSource\_FileOrURL\_StartTime

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the start time of the [VideoSource\\_FileOrUrl](#) video clip.

### Declaration

**property** VideoSource\_FileOrURL\_StartTime: LargeInteger **read** GetVideoSource\_FileOrURL\_StartTime  
**write** SetVideoSource\_FileOrURL\_StartTime **default** DEF\_VideoSource\_FileOrURL\_StartTime;

```
__property __int64 VideoSource_FileOrURL_StartTime=read=GetVideoSource_FileOrURL_StartTime,
write=SetVideoSource_FileOrURL_StartTime, default=- 1
```

Property VideoSource\_FileOrURL\_StartTime As Double

### Description

Specifies the start time of the [VideoSource\\_FileOrUrl](#) video clip when [VideoSource](#) = vs\_VideoFileOrUrl. The start time is expressed in 100ns units (e.g. 20000000 = 2 seconds).

When it is set to **-1** the video clip starts from the beginning.

#### See Also

[VideoSource\\_FileOrURL\\_VideoSource\\_FileOrURL\\_StopTime](#)

---

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

---

## VideoSource\_FileOrURL\_StopTime

### TVideoGrabber.VideoSource\_FileOrURL\_StopTime

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Specifies the stop time of the [VideoSource\\_FileOrUrl](#) video clip.

#### Declaration

**property** VideoSource\_FileOrURL\_StopTime: LargeInteger **read** GetVideoSource\_FileOrURL\_StopTime  
**write** SetVideoSource\_FileOrURL\_StopTime **default** DEF\_VideoSource\_FileOrURL\_StopTime;

\_\_property \_\_int64 VideoSource\_FileOrURL\_StopTime=read=GetVideoSource\_FileOrURL\_StopTime,  
write=SetVideoSource\_FileOrURL\_StopTime, **default**=- 1

Property VideoSource\_FileOrURL\_StopTime As Double

#### Description

Specifies the stop time of the [VideoSource\\_FileOrUrl](#) video clip when [VideoSource](#) = vs\_VideoFileOrUrl.  
The stop time is expressed in 100ns units (e.g. 20000000 = 2 seconds).  
When it is set to **-1** the video clip will play until the end.

#### See Also

[VideoSource\\_FileOrURL\\_VideoSource\\_FileOrURL\\_StartTime](#)

---

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

---

## VideoSources

### TVideoGrabber.VideoSources

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video sources available.

#### Declaration

**property** VideoSources: **string** **read** GetVideoSources;

\_\_property wchar\_t \*VideoSources=read=GetVideoSources

Property VideoSources As String

#### Description

List of the video sources available.  
Used to select the [VideoSource](#), that is an index in this list.

#### See Also

[Video sources supported for preview and recording\\_VideoSource\\_VideoSourcesCount\\_TVideoSource](#)

---

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

---

## VideoSourcesCount

### TVideoGrabber.VideoSourcesCount

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Number of items in the [VideoSources](#) list.

#### Declaration

**property** VideoSourcesCount: LongInt **read** GetVideoSourcesCount;

\_\_property **int** VideoSourcesCount=read=GetVideoSourcesCount, **nodefault**

Property VideoSourcesCount As Long

#### Description

Returns the number of items in the [VideoSources](#) list.

#### See Also

[Video sources supported for preview and recording](#) [VideoSource](#) [VideoSources](#) [TVideoSource](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Encrypted, Password-Protected PDFs](#)

## VideoStreamNumber

### TVideoGrabber.VideoStreamNumber

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Let choose the video stream

#### Declaration

**property** VideoStreamNumber: LongInt **read** GetVideoStreamNumber **write** GetVideoStreamNumber **default** DEF\_VideoStreamNumber;

\_\_property **int** VideoStreamNumber=read=GetVideoStreamNumber, write=GetVideoStreamNumber, **default**=DEF\_VideoStreamNumber

#### Description

This property let choose the video stream when clips have multiple video streams.  
The stream number must be in the 0..n-1 range (default = -1)

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

## VideoSubtype

### TVideoGrabber.VideoSubtype

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Properties](#)

Selects a video subtype.

#### Declaration

**property** VideoSubtype: LongInt **read** GetVideoSubtype **write** SetVideoSubtype;

\_\_property **int** VideoSubtype=read=GetVideoSubtype, write=SetVideoSubtype, **nodefault**

Property VideoSubtype As Long

## Description

Used to select a video subtype in the [VideoSubtypes](#) list.  
See [VideoSubtypes](#) .

### Important:

The current video subtype index is not the same from a video capture device to the other. It is reloaded when the video capture device is selected by assigning the [VideoDevice](#) property.

- any control displaying this value should re-read it from the [OnVideoDeviceSelected](#) event (that occurs when the video device is selected)
- be sure that the video capture device has been selected (by assigning the [VideoDevice](#) property) before setting this property

See [device-dependent properties](#)

### See Also

[VideoSubtypeIndex](#) [VideoSubtypes](#) [VideoSubtypesCount](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

## VideoSubtypes

### TVideoGrabber.VideoSubtypes

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

List of the video subtypes available on the current video capture device

### Declaration

**property** VideoSubtypes: **string read** GetVideoSubtypes;

\_\_property wchar\_t \*VideoSubtypes=read=GetVideoSubtypes

Property VideoSubtypes As String

### Description

Used to retrieve a string that contains the list of the video subtypes available on the current video capture device.

This list is updated when the [OnVideoDeviceSelected](#) event occurs (when a video capture device is selected with [VideoDevice](#) ).

This list can be assigned to list based controls. E.g.:

*ComboBox1.Items.Text := VideoGrabber1.VideoSubtypes;*

*ComboBox1.ItemIndex := VideoGrabber1.VideoSubtype;*

Note: as this list is not the same from a video capture device to the other, any code using this value should re-read it from the [OnVideoDeviceSelected](#) event (when a [video capture device](#) is selected).

### See Also

[VideoSubtype](#) [VideoSubtypeIndex](#) [VideoSubtypesCount](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## VideoSubtypesCount

### TVideoGrabber.VideoSubtypesCount

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Properties](#)

Number of video subtypes.

#### Declaration

**property** VideoSubtypesCount: LongInt **read** GetVideoSubtypesCount;

\_\_property int VideoSubtypesCount=read=GetVideoSubtypesCount, **nodefault**

Property VideoSubtypesCount As Long

#### Description

Number of video subtypes in the [VideoSubtypes](#) list for the current video capture device.

#### See Also

[VideoSubtype](#) [VideoSubtypeIndex](#) [VideoSubtypes](#)

Created with the Standard Edition of HelpNDoc: [Produce online help for Qt applications](#)

## VideoVisibleWhenStopped

### TVideoGrabber.VideoVisibleWhenStopped

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Prevent the video window to be closed when the playback or preview ends.

#### Declaration

**property** VideoVisibleWhenStopped: Boolean **read** GetVideoVisibleWhenStopped **write** SetVideoVisibleWhenStopped **default** DEF\_VideoVisibleWhenStopped;

\_\_property bool VideoVisibleWhenStopped==GetVideoVisibleWhenStopped,  
write=SetVideoVisibleWhenStopped, **default**=0;

property VideoVisibleWhenStopped as Boolean

#### Description

When enabled, keeps the video window active after the the playback or preview ends.  
The last video frame remains displayed until [StartPreview](#), [StartRecording](#) or [OpenPlayer](#) is invoked.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#)  
[Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#)  
[Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#)  
[Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#)  
[Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#)  
[Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#)  
[DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#)  
[DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#)  
[DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#)  
[DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#)  
[DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Visible](#) [DualDisplay](#) [Width](#)  
[IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#)  
[SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#)  
[VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

## VideoWidth

### TVideoGrabber.VideoWidth

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Retrieves the width of the native video source, in pixels.

### Declaration

**property** VideoWidth: LongInt **read** GetVideoWidth;

**\_\_property** int VideoWidth=read=GetVideoWidth, **nodefault**

Property VideoWidth As Long

### Description

Used to retrieve the width of the native video source, in pixels.

### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Free PDF documentation generator](#)

## [VideoWidth\\_PreferedAspectRatio](#)

### [TVideoGrabber.VideoWidth\\_PreferedAspectRatio](#)

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Properties](#)

Retrieves the preferred display width of the video clip, in pixels.

### Declaration

**property** VideoWidth\_PreferedAspectRatio: LongInt **read** GetVideoWidth\_PreferedAspectRatio;

**\_\_property** int VideoWidth\_PreferedAspectRatio=read=GetVideoWidth\_PreferedAspectRatio, **nodefault**

Property VideoWidth\_PreferedAspectRatio As Long

### Description

Used to retrieve the preferred display width of the video clip, in pixels.

It can be different of the [VideoWidth](#) property if the video clip has a display aspect ratio (DAR) different of the source aspect ratio (SAR)

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TVideoRenderer](#) [TOnPlayerBufferingData](#) [AdjustPixelAspectRatio](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [IsVideoPortAvailable](#) [Last\\_Clip](#) [Played](#) [Monitor](#) [Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [MP4NeedsReindexing](#)

[OnLeavingFullScreen](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SetParentWindow](#) [SetWindowTransparency](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#)

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

## Visible

### TVideoGrabber.Visible

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Used to make the component visible / not visible.

#### Declaration

**property** Visible;

\_\_property Visible

#### Description

Enable / disable the Visible property to make the component visible or not visible.

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#)

Created with the Standard Edition of HelpNDoc: [Free EBook and documentation generator](#)

## VUMeter

### TVideoGrabber.VUMeter

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enable the audio VU-meters.

#### Declaration

**property** VUMeter: TVUMeter **read** GetVUMeter **write** SetVUMeter **default** DEF\_VUMeter;

\_\_property TVUMeter VUMeter=read=GetVUMeter, write=SetVUMeter, **default**=0

Property VUMeter As TxVUMeter

#### Description

Used to enable the audio VU-Meters.

Possible values:

vu\_Disabled : VU-Meters disabled  
vu\_Analog : analog VU-Meters



vu\_Bargraph : bargraph VU-Meters

See the "[Audio levels and VU-Meters](#)" chapter, that explains how to activate the VU-Meters.

#### See Also

[TVUMeter](#) [TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## WebcamStillCaptureButton

### TVideoGrabber.WebcamStillCaptureButton

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Enable the frame capture by using the webcam's still capture button

#### Declaration

**property** WebcamStillCaptureButton: TWebcamStillCaptureButton **read** GetWebcamStillCaptureButton  
**write** SetWebcamStillCaptureButton **default** DEF\_WebcamStillCaptureButton;

**\_\_property** TWebcamStillCaptureButton WebcamStillCaptureButton=read=GetWebcamStillCaptureButton,  
write=SetWebcamStillCaptureButton, **default**=0

Property WebcamStillCaptureButton As TxWebcamStillCaptureButton

#### Description

Used to enable the frame capture when the webcam includes a still capture button.

The type of the property is [TWebcamStillCaptureButton](#) , possible values:

#### wb\_Disabled:

the still capture button is disabled

#### wb\_Enabled :

the still capture button is enabled, when pressed a frame will be captured.

The captured frame will be returned by the [OnFrameCaptureCompleted](#) event, according to a memory bitmap, a BMP file or a JPEG file according to the [BurstType](#) property (the filename structure depends on the [AutoFileName](#) property)

#### See Also

[WDM drivers](#) [TAutoFileName](#) [TFrameCaptureDest](#) [TFrameGrabberRGBFormat](#)  
[TOnFrameCaptureCompleted](#) [AutoConnectRelatedPins](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#)  
[CaptureFrameSyncTo](#) [CaptureFrameTo](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#)  
[FixFlickerOrBlackCapture](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#)  
[FrameCaptureZoomSize](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#) [FrameGrabberRGBFormat](#)  
[FramerateDivider](#) [GetFrameInfo](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#)  
[GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [InFrameProgressEvent](#)  
[IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#)  
[IsWDMVideoDriver](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#)  
[Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#)  
[OnFrameBitmapEventSynchronise](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#)  
[RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [SetFrameCaptureBounds](#) [ShowDialog](#)  
[StoragePath](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#)  
[VideoDevicesCount](#) [VideoDevicesId](#)

Created with the Standard Edition of HelpNDoc: [Quickly and Easily Convert Your Word Document to an ePub or Kindle eBook](#)

## ZoomCoeff

### TVideoGrabber.ZoomCoeff

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Zooming coefficient X 1000

#### Declaration

**property** ZoomCoeff: LongInt **read** GetZoomCoeff **write** SetZoomCoeff;

\_\_property **int** ZoomCoeff=read=GetZoomCoeff, write=SetZoomCoeff, **nodefault**

Property ZoomCoeff As Long

#### Description

Zooming coefficient, expressed as per thousand.  
e.g:

1000 = no zooming (default value)  
1500 = 1.5x zoom  
2000 = 2x zoom  
etc...

The [frame grabber](#) must be enabled to use this feature.

Created with the Standard Edition of HelpNDoc: [Experience the Power and Simplicity of HelpNDoc's User Interface](#)

## ZoomXCenter

### TVideoGrabber.ZoomXCenter

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

X center position of the zooming

#### Declaration

**property** ZoomXCenter: LongInt **read** GetZoomXCenter **write** SetZoomXCenter;

\_\_property **int** ZoomXCenter=read=GetZoomXCenter, write=SetZoomXCenter, **nodefault**

Property ZoomXCenter As Long

#### Description

Secifies the vertical center position when the [zooming](#) is used, expressed in pixels.

0 means the vertical center of the video frame (default value)  
- 50 means that the vertical center of the zooming is shifted of 50 pixels in the left direction  
+ 50 means that the vertical center of the zooming is shifted of 50 pixels in the right direction

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## ZoomYCenter

### TVideoGrabber.ZoomYCenter

[Prev](#)

[TVideoGrabber](#) [Properties](#)

Y center position of the zooming

#### Declaration

**property** ZoomYCenter: LongInt **read** GetZoomYCenter **write** SetZoomYCenter;

\_\_property **int** ZoomYCenter=read=GetZoomYCenter, write=SetZoomYCenter, **nodefault**

Property ZoomYCenter As Long

#### Description

Secifies the horizontal center position when the [zooming](#) is used, expressed in pixels.

0 means the horizontal center of the video frame (default value)

- 50 means that the vertical center of the zooming is shifted of 50 pixels in the top direction

+ 50 means that the vertical center of the zooming is shifted of 50 pixels in the bottom direction

---

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

## Methods

### TVideoGrabber Methods

#### TVideoGrabber

#### Public

[About](#)

[AnalogVideoStandardIndex](#)

[ASFStreaming\\_GetAuthorizationList](#)

[ASFStreaming\\_GetConnectedClients](#)

[ASFStreaming\\_GetConnectedClientsCount](#)

[ASFStreaming\\_ResetAuthorizations](#)

[ASFStreaming\\_SetAuthorization](#)

[AssociateMultiplexedSlave](#)

[AudioCompressorIndex](#)

[AudioDeviceIndex](#)

[AudioInputIndex](#)

[AudioRendererIndex](#)

[AVIDuration](#)

[AVIHeaderInfo](#)

[AVIInfo](#)

[AVIInfo2](#)

[CameraControlAuto](#)

[CameraControlDefault](#)

[CameraControlMax](#)

[CameraControlMin](#)

[CameraControlStep](#)

[CameraControlValue](#)

[Cancel](#)

[CanProcessMessages](#)

[CaptureFrameRenderedTo](#)

[CaptureFrameSyncTo](#)

[CaptureFrameTo](#)

[ClearHeaderAttributes](#)

[ClosePlayer](#)

[Create](#)

[CreatePreallocCapFile](#)

[Decrypt\\_File](#)

[Destroy](#)

[Display\\_SetLocation](#)

[DrawBitmapOverFrame](#)

[DualDisplay\\_SetLocation](#)

[DVDInfo](#)

[EnableMultiplexedInput](#)  
[EnableThreadMode](#)  
[Encoder\\_CloseOutputFile](#)  
[Encoder\\_GetInt](#)  
[Encoder\\_NewOutputFile](#)  
[Encoder\\_Pause](#)  
[Encoder\\_Resume](#)  
[Encoder\\_SetInt](#)  
[Encoder\\_SetStr](#)  
[Encoders\\_CreateInstanceForRecording](#)  
[Encoders\\_CreateInstanceForStreaming](#)  
[Encoders\\_RemoveAllInstances](#)  
[Encoders\\_RemoveInstance](#)  
[Encrypt\\_File](#)  
[EnumerateWindows](#)  
[Facebook\\_GoLive\\_ReturnStreamURL](#)  
[FastForwardPlayer](#)  
[FindIndexInListByName](#)  
[GetAudioCodec](#)  
[GetCameraExposure](#)  
[GetCameraExposureAsString](#)  
[GetDisplayActive](#)  
[GetDisplayAlphaBlendEnabled](#)  
[GetDisplayAlphaBlendValue](#)  
[GetDisplayAspectRatio](#)  
[GetDisplayAutoSize](#)  
[GetDisplayEmbedded](#)  
[GetDisplayFullScreen](#)  
[GetDisplayHeight](#)  
[GetDisplayLeft](#)  
[GetDisplayMonitor](#)  
[GetDisplayMouseMovesWindow](#)  
[GetDisplayPanScanRatio](#)  
[GetDisplayStayOnTop](#)  
[GetDisplayTop](#)  
[GetDisplayTransparentColorEnabled](#)  
[GetDisplayTransparentColorValue](#)  
[GetDisplayVideoHeight](#)  
[GetDisplayVideoPortEnabled](#)  
[GetDisplayVideoWidth](#)  
[GetDisplayVideoWindowHandle](#)  
[GetDisplayVisible](#)  
[GetDisplayWidth](#)  
[GetFrameInfo](#)  
[GetFrameInfoString](#)  
[GetFWCam1394](#)  
[GetFWCam1394List](#)  
[GetImageOverlay\\_AlphaBlend](#)  
[GetImageOverlay\\_AlphaBlendValue](#)  
[GetImageOverlay\\_ChromaKey](#)  
[GetImageOverlay\\_ChromaKeyLeewayPercent](#)  
[GetImageOverlay\\_ChromaKeyRGBColor](#)  
[GetImageOverlay\\_Enabled](#)  
[GetImageOverlay\\_Height](#)  
[GetImageOverlay\\_LeftLocation](#)  
[GetImageOverlay\\_RotationAngle](#)  
[GetImageOverlay\\_StretchToVideoSize](#)  
[GetImageOverlay\\_TargetDisplay](#)  
[GetImageOverlay\\_TopLocation](#)  
[GetImageOverlay\\_Transparent](#)  
[GetImageOverlay\\_TransparentColorValue](#)  
[GetImageOverlay\\_UseTransparentColor](#)  
[GetImageOverlay\\_Width](#)

[GetItemNameFromList](#)  
[GetLastAverageStreamValue](#)  
[GetLastErrorMessage](#)  
[GetLastFrameAsHBITMAP](#)  
[GetLastFrameAsTBitmap](#)  
[GetLastFrameBitmapBits](#)  
[GetLastFrameBitmapBits2](#)  
[GetLogString](#)  
[GetMiscDeviceControl](#)  
[GetNearestVideoHeight](#)  
[GetNearestVideoSize](#)  
[GetNearestVideoWidth](#)  
[GetPixelsDistance](#)  
[GetPlaylist](#)  
[GetRGBPixelAt](#)  
[GetTextOverlay\\_Align](#)  
[GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#)  
[GetTextOverlay\\_BkColor](#)  
[GetTextOverlay\\_Enabled](#)  
[GetTextOverlay\\_Font](#)  
[GetTextOverlay\\_GradientColor](#)  
[GetTextOverlay\\_GradientMode](#)  
[GetTextOverlay\\_HighResFont](#)  
[GetTextOverlay\\_Left](#)  
[GetTextOverlay\\_Right](#)  
[GetTextOverlay\\_Scrolling](#)  
[GetTextOverlay\\_ScrollingSpeed](#)  
[GetTextOverlay\\_Shadow](#)  
[GetTextOverlay\\_ShadowColor](#)  
[GetTextOverlay\\_ShadowDirection](#)  
[GetTextOverlay\\_String](#)  
[GetTextOverlay\\_TargetDisplay](#)  
[GetTextOverlay\\_Top](#)  
[GetTextOverlay\\_Transparent](#)  
[GetTranslatedCoordinates](#)  
[GetTVChannelInfo](#)  
[GetVideoCodec](#)  
[GetVideoCompressionSettings](#)  
[GetVideoControlMode](#)  
[GetVideoHeightFromIndex](#)  
[GetVideoSizeFromIndex](#)  
[GetVideoWidthFromIndex](#)  
[GetVMR9ImageAdjustmentBounds](#)  
[GetVUMeterSetting](#)  
[GraphState](#)  
[IsAudioDeviceConnected](#)  
[IsAudioRendererConnected](#)  
[IsCameraControlSettingAvailable](#)  
[IsDialogAvailable](#)  
[IsDVDDevice](#)  
[IsPlaylistActive](#)  
[IsURLResponding](#)  
[IsURLVideoStreamAvailable](#)  
[IsVideoControlModeAvailable](#)  
[IsVideoDeviceConnected](#)  
[IsVideoQualitySettingAvailable](#)  
[IsVideoSignalDetected](#)  
[IsVMR9ImageAdjustmentAvailable](#)  
[LoadCompressorSettingsFromDataString](#)  
[LoadCompressorSettingsFromTextFile](#)  
[MixAudioSamples](#)  
[Mixer\\_Activation](#)

[Mixer\\_AddToMixer](#)  
[Mixer\\_RemoveFromMixer](#)  
[Mixer\\_SetOverlayRoundedCorner](#)  
[Mixer\\_SetupPIPFromSource](#)  
[Monitor\\_Primary\\_Index](#)  
[MonitorBounds](#)  
[MonitorsCount](#)  
[MotionDetector\\_CellColorIntensity](#)  
[MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_EnumGridDialogControls](#)  
[MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion](#)  
[MotionDetector\\_GetCellLocation](#)  
[MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize](#)  
[MotionDetector\\_GlobalColorIntensity](#)  
[MotionDetector\\_GloballyIncOrDecSensitivity](#)  
[MotionDetector\\_Reset](#)  
[MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#)  
[MotionDetector\\_SetGridSize](#)  
[MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_TriggerNow](#)  
[MotionDetector\\_UseThisReferenceSample](#)  
[MP4NeedsReindexing](#)  
[MPEGProgramSetting](#)  
[MultipurposeEncoder\\_QuickConfigure\\_UDPStreaming\\_H264](#)  
[MultipurposeEncoder\\_ReindexClip](#)  
[NotifyPlayerTrackbarAction](#)  
[ONVIF\\_GetStr](#)  
[ONVIF\\_SetStr](#)  
[ONVIFCancelDiscovery](#)  
[ONVIFDeviceInfo](#)  
[ONVIFDiscoverCameras\\_IPRange](#)  
[ONVIFDiscoverCameras\\_Multicast](#)  
[ONVIFEnumCamerasDiscovered](#)  
[ONVIFPTZGetLimits](#)  
[ONVIFPTZGetPosition](#)  
[ONVIFPTZPreset](#)  
[ONVIFPTZSendAuxiliaryCommand](#)  
[ONVIFPTZSetPosition](#)  
[ONVIFPTZStartMove](#)  
[ONVIFPTZStopMove](#)  
[ONVIFSnapShot](#)  
[OpenDVD](#)  
[OpenPlayer](#)  
[OpenPlayerAtFramePositions](#)  
[OpenPlayerAtTimePositions](#)  
[OpenURLAsyncStatus](#)  
[PausePlayer](#)  
[PausePreview](#)  
[PauseRecording](#)  
[PlayerFrameStep](#)  
[Playlist](#)  
[PointGreyConfig](#)  
[PutMiscDeviceControl](#)  
[RecordingKBytesWrittenToDisk](#)  
[RecordToNewFileNow](#)  
[RefreshDevicesAndCompressorsLists](#)  
[RefreshPlayerOverlays](#)  
[ResetVideoDeviceSettings](#)  
[ResumePreview](#)  
[ResumeRecording](#)

[RetrieveInitialXYAfterRotation](#)  
[RewindPlayer](#)  
[RunPlayer](#)  
[RunPlayerBackwards](#)  
[SaveCompressorSettingsToDataString](#)  
[SaveCompressorSettingsToTextFile](#)  
[ScreenRecordingUsingCoordinates](#)  
[SendCameraCommand](#)  
[SendDVCommand](#)  
[SendImageToVideoFromBitmaps](#)  
[SendImageToVideoFromBitmaps2](#)  
[SendIPCameraCommand](#)  
[SetAudioRendererAdditional](#)  
[SetAuthentication](#)  
[SetAVIMuxConfig](#)  
[SetCameraControl](#)  
[SetCameraExposure](#)  
[SetDecryptionKey](#)  
[SetDisplayActive](#)  
[SetDisplayAlphaBlendEnabled](#)  
[SetDisplayAlphaBlendValue](#)  
[SetDisplayAspectRatio](#)  
[SetDisplayAutoSize](#)  
[SetDisplayEmbedded](#)  
[SetDisplayFullScreen](#)  
[SetDisplayHeight](#)  
[SetDisplayLeft](#)  
[SetDisplayLocation](#)  
[SetDisplayMonitor](#)  
[SetDisplayMouseMovesWindow](#)  
[SetDisplayPanScanRatio](#)  
[SetDisplayParent](#)  
[SetDisplayStayOnTop](#)  
[SetDisplayTop](#)  
[SetDisplayTransparentColorEnabled](#)  
[SetDisplayTransparentColorValue](#)  
[SetDisplayVideoPortEnabled](#)  
[SetDisplayVisible](#)  
[SetDisplayWidth](#)  
[SetEncryptionKey](#)  
[SetFFmpegAudioFilter](#)  
[SetFFmpegFilter](#)  
[SetFrameCaptureBounds](#)  
[SetFWCam1394](#)  
[SetHeaderAttribute](#)  
[SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#)  
[SetImageOverlay\\_Attributes](#)  
[SetImageOverlay\\_Attributes2](#)  
[SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#)  
[SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#)  
[SetImageOverlay\\_Height](#)  
[SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#)  
[SetImageOverlay\\_StretchToVideoSize](#)  
[SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#)  
[SetImageOverlay\\_Transparent](#)  
[SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#)  
[SetImageOverlay\\_Width](#)



[SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromBMPFile2](#)  
[SetImageOverlayFromHBitmap](#)  
[SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#)  
[SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromJPEGFile2](#)  
[SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#)  
[SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#)  
[SetIPCameraSetting](#)  
[SetLocation](#)  
[SetLogoFromBMPFile](#)  
[SetLogoFromHBitmap](#)  
[SetLogoFromJPEGFile](#)  
[SetLogoFromTBitmap](#)  
[SetLogoFromTImage](#)  
[SetMultiplexerFilterByName](#)  
[SetParentWindow](#)  
[SetTextOverlay\\_Align](#)  
[SetTextOverlay\\_AlphaBlend](#)  
[SetTextOverlay\\_AlphaBlendValue](#)  
[SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#)  
[SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#)  
[SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#)  
[SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#)  
[SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#)  
[SetTextOverlay\\_ScrollingSpeed](#)  
[SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#)  
[SetTextOverlay\\_ShadowDirection](#)  
[SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#)  
[SetTextOverlay\\_Top](#)  
[SetTextOverlay\\_Transparent](#)  
[SetVideoCompressionDefaults](#)  
[SetVideoCompressionSettings](#)  
[SetVideoControlMode](#)  
[SetVideoControlMode2](#)  
[SetVideoQuality](#)  
[SetVMR9ImageAdjustmentValue](#)  
[SetVuMeter\\_Enabled](#)  
[SetVUMeterSetting](#)  
[SetWindowRecordingByHandle](#)  
[SetWindowRecordingByName](#)  
[SetWindowTransparency](#)  
[ShapeOverlayList](#)  
[ShowDebugWindow](#)  
[ShowDialog](#)  
[StartAudioRecording](#)  
[StartAudioRendering](#)  
[StartPreview](#)  
[StartRecording](#)  
[StartReencoding](#)  
[StartSynchronized](#)

[Stop](#)  
[StopPlayer](#)  
[StopPreview](#)  
[StopRecording](#)  
[StopReencoding](#)  
[StreamInterface PushData](#)  
[TextOverlay CreateCustomFont](#)  
[TextOverlay CreateCustomFont2](#)  
[ThirdPartyFilter AddToList](#)  
[ThirdPartyFilter ClearList](#)  
[ThirdPartyFilter Enable](#)  
[ThirdPartyFilter RemoveFromList](#)  
[ThirdPartyFilter ShowDialog](#)  
[TVClearFrequencyOverrides](#)  
[TVGetMinMaxChannels](#)  
[TVSetChannelFrequencyOverride](#)  
[TVStartAutoScan](#)  
[TVStopAutoScan](#)  
[UseNearestVideoSize](#)  
[v360 AddYawPitchRoll](#)  
[v360 GetAngle](#)  
[v360 GetYawPitchRoll](#)  
[v360 ResetAnglesToDefault](#)  
[v360 SetAngle](#)  
[v360 SetInterpolation](#)  
[v360 SetProjection](#)  
[v360 SetStereoFormat](#)  
[v360 SetTranspose](#)  
[v360 SetYawPitchRoll](#)  
[VideoCompressorIndex](#)  
[VideoDeviceIndex](#)  
[VideoDeviceIndexFromId](#)  
[VideoFormatIndex](#)  
[VideoFromImages CreateSetOfBitmaps](#)  
[VideoInputIndex](#)  
[VideoQualityAuto](#)  
[VideoQualityDefault](#)  
[VideoQualityMax](#)  
[VideoQualityMin](#)  
[VideoQualityStep](#)  
[VideoQualityValue](#)  
[VideoSizeIndex](#)  
[VideoSubtypeIndex](#)  
[WriteScriptCommand](#)

---

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

---

## About

### TVideoGrabber.About

[Next](#)

[TVideoGrabber](#)   [Methods](#)

Shows the current version of TVideoGrabber

#### Declaration

**procedure** About;

**void** \_\_fastcall About(**void**)

Sub About()

## Description

Invokes a dialog that displays the current version of TVideoGrabber.

Created with the Standard Edition of HelpNDoc: [Experience the Power and Simplicity of HelpNDoc's User Interface](#)

## AnalogVideoStandardIndex

### TVideoGrabber.AnalogVideoStandardIndex

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Returns the index of a given analog video standard in the [AnalogVideoStandards](#) list.

#### Declaration

**function** AnalogVideoStandardIndex(Value: **String**): LongInt;

**int** \_\_fastcall AnalogVideoStandardIndex(wchar\_t \*Value)

Function AnalogVideoStandardIndex(param1 As String) As Long

#### Description

Used to retrieve the index of a given analog video standard in the [AnalogVideoStandards](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.AnalogVideoStandard := AnalogVideoStandardIndex ('PAL B'); // selects the
VideoGrabber1.AnalogVideoStandard := AnalogVideoStandardIndex ('PAL*'); // selects the
VideoGrabber1.AnalogVideoStandard := AnalogVideoStandardIndex (*PAL*'); // selects the
```

#### See Also

[Analog Video standards](#) [AnalogVideoStandard](#) [AnalogVideoStandards](#) [AnalogVideoStandardsCount](#)  
[IsAnalogVideoDecoderAvailable](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Documentation Process with HelpNDoc's Project Analyzer](#)

## ASFStreaming\_GetAuthorizationList

### TVideoGrabber.ASFStreaming\_GetAuthorizationList

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Returns the list of the ASF streaming authorizations.

#### Declaration

**function** ASFStreaming\_GetAuthorizationList: **string**;

wchar\_t \* \_\_fastcall ASFStreaming\_GetAuthorizationList()

Function ASFStreaming\_GetAuthorizationList as String

#### Description

Returns the list of the current ASF streaming authorizations (allowed or blocked).  
E.g.:

```
block: 192.168.0.53/255.255.255.255
block: 192.168.0.194/255.255.255.255
```

### See Also

[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetConnectedClients](#) [ASFStreaming\\_GetConnectedClientsCount](#)  
[ASFStreaming\\_ResetAuthorizations](#) [ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)

Created with the Standard Edition of HelpNDoc: Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc

## ASFStreaming\_GetConnectedClients

### TVideoGrabber.ASFStreaming\_GetConnectedClients

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the list of the clients currently connected to the ASF streaming.

#### Declaration

**function** ASFStreaming\_GetConnectedClients: **string**;

wchar\_t \* \_\_fastcall ASFStreaming\_GetConnectedClients()

Function ASFStreaming\_GetConnectedClients as String

#### Description

Returns the list of the clients currently connected to the ASF streaming (IP address and port).  
 E.g.:

```
192.168.0.121:4180
192.168.0.51:4685
```

### See Also

[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClientsCount](#)  
[ASFStreaming\\_ResetAuthorizations](#) [ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)

Created with the Standard Edition of HelpNDoc: Streamline Your CHM Help File Creation with HelpNDoc

## ASFStreaming\_GetConnectedClientsCount

### TVideoGrabber.ASFStreaming\_GetConnectedClientsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the number of clients currently connected to the ASF streaming.

#### Declaration

**function** ASFStreaming\_GetConnectedClientsCount: LongInt;

**int** \_\_fastcall ASFStreaming\_GetConnectedClientsCount(**void**)

Function ASFStreaming\_GetConnectedClientsCount as Long

### Description

Returns the number of clients currently connected to the ASF streaming.

### See Also

[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#)  
[ASFStreaming\\_ResetAuthorizations](#) [ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)

Created with the Standard Edition of HelpNDoc: [Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion](#)

## ASFStreaming\_ResetAuthorizations

### TVideoGrabber.ASFStreaming\_ResetAuthorizations

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Erase all the current ASF streaming authorizations.

### Declaration

**function** ASFStreaming\_ResetAuthorizations: Boolean;

**bool** \_\_fastcall ASFStreaming\_ResetAuthorizations(**void**)

Public Overridable Sub ASFStreaming\_ResetAuthorizations()

### Description

Used to erase the current ASF streaming authorizations that have been set by invoking [ASFStreaming\\_SetAuthorization](#).

### See Also

[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#)  
[ASFStreaming\\_GetConnectedClientsCount](#) [ASFStreaming\\_SetAuthorization](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Encrypted, Password-Protected PDFs](#)

## ASFStreaming\_SetAuthorization

### TVideoGrabber.ASFStreaming\_SetAuthorization

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Prevents an IP address to connect to the ASF streaming.

### Declaration

**function** ASFStreaming\_SetAuthorization(Allowed: Boolean; IP: **String**; Mask: **String**): Boolean;

**bool** \_\_fastcall ASFStreaming\_SetAuthorization(**bool** Allowed, wchar\_t \*IP, wchar\_t \*Mask)

Function ASFStreaming\_SetAuthorization (Allowed as Boolean, IP as String, Mask as String) as Boolean

### Description

Used to block the ASF streaming for the specified IP address.

E.g:

- to block the 192.168.0.50 ip address:

```
ASFStreaming_SetAuthorization (false, '192.168.0.51', '255.255.255.255');
```

- to block all the IP addresses on 192.168.0.xxx:

```
ASFStreaming_SetAuthorization (false, '192.168.0.0', '255.255.255.0');
```

### See Also

[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#)  
[ASFStreaming\\_GetAuthorizationList](#) [ASFStreaming\\_GetConnectedClients](#)  
[ASFStreaming\\_GetConnectedClientsCount](#) [ASFStreaming\\_ResetAuthorizations](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## AssociateMultiplexedSlave

### TVideoGrabber.AssociateMultiplexedSlave

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Associates a slave to a master component.

### Declaration

**function** AssociateMultiplexedSlave(InputNumber: LongInt; SlaveUniqueID: LongInt): Boolean;

**bool** \_\_fastcall AssociateMultiplexedSlave (**int** InputNumber, **int** SlaveUniqueID)

Function AssociateMultiplexedSlave(InputNumber As Long, SlaveUniqueID As Long) As Boolean

### Description

Used in slave/master [multiplexed mode](#) to associate a slave TVideoGrabber component to a master TVideoGrabber component.

### See Also

[TMultiplexedRole](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#) [MultiplexedRole](#)  
[MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## AudioCompressorIndex

### TVideoGrabber.AudioCompressorIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given compressor in the [AudioCompressors](#) list.

#### Declaration

**function** AudioCompressorIndex(Value: **String**): LongInt;

**int** \_\_fastcall AudioCompressorIndex(wchar\_t \*Value)

Function AudioCompressorIndex(Value As String) As Long

#### Description

Used to retrieve the index of a given compressor in the [AudioCompressors](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.AudioCompressor := AudioCompressorIndex ('MPEG Layer-3'); // selects the "MPEG Layer-3" compressor
VideoGrabber1.AudioCompressor := AudioCompressorIndex ('MPEG*'); // selects the first MPEG compressor
VideoGrabber1.AudioCompressor := AudioCompressorIndex ('*MPEG*'); // selects the first MPEG compressor
```

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Publish Your Word Document as an eBook](#)

## AudioDeviceIndex

### TVideoGrabber.AudioDeviceIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given audio device in the [AudioDevices](#) list.

#### Declaration

**function** AudioDeviceIndex(Value: **String**): LongInt;

**int** \_\_fastcall AudioDeviceIndex(wchar\_t \*Value)

Function AudioDeviceIndex(Value As String) As Long

#### Description

Used to retrieve the index of a given audio device in the [AudioDevices](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.AudioDevice := AudioDeviceIndex ('Hauppauge WinTV USB Pro Audio (WDM)'); // selects the "Hauppauge WinTV USB Pro Audio (WDM)" device
VideoGrabber1.AudioDevice := AudioDeviceIndex ('Hauppauge*'); // selects the first Hauppauge device
VideoGrabber1.AudioDevice := AudioDeviceIndex ('*Hauppauge*'); // selects the first Hauppauge device
```

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#)



[AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#)  
[IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#)  
[IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#)  
[SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## AudioInputIndex

### TVideoGrabber.AudioInputIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given audio input in the [AudioInputs](#) list.

#### Declaration

**function** AudioInputIndex(Value: **String**): LongInt;

**int** \_\_fastcall AudioInputIndex(wchar\_t \*Value)

Function AudioInputIndex(Value As String) As Long

#### Description

Used to retrieve the index of a given audio input in the [AudioInputs](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.AudioInput := AudioInputIndex ('Audio CD'); // selects the "Audio CD" audio
VideoGrabber1.AudioInput := AudioInputIndex ('CD*');      // selects the 1st audio input
VideoGrabber1.AudioInput := AudioInputIndex ('*CD*');     // selects the 1st audio input
```

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#)  
[AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#)  
[AudioInput](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#)  
[IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#)  
[IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#)  
[SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

## AudioRendererIndex

### TVideoGrabber.AudioRendererIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieve the index of a given audio renderer in the [AudioRenderers](#) list.

#### Declaration

**function** AudioRendererIndex (Value: **string**): LongInt;

**int** \_\_fastcall AudioRendererIndex(wchar\_t \*Value);

Function AudioRendererIndex(Value As String) As Long

#### Description

Used to retrieve the index of a given audio renderer in the [AudioRenderers](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.AudioRenderer := AudioRendererIndex ('DirectSound: Realtek AC97 Audio');
VideoGrabber1.AudioRenderer := AudioRendererIndex ('*Realtek AC97 Audio');
VideoGrabber1.AudioRenderer := AudioRendererIndex ('*AC97*');
```

#### See Also

[IsAudioRendererConnected](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## AVIDuration

### TVideoGrabber.AVIDuration

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the duration of the AVI file specified.

#### Declaration

**function** AVIDuration(AVIFile: **String**; **out** Duration: LargeInteger; **out** FrameCount: LargeInteger): Boolean;

**bool** \_\_fastcall AVIDuration(wchar\_t \*AVIFile, \_\_int64 &Duration, \_\_int64 &FrameCount)

Function AVIDuration(AVIFile As String, AVIDuration, AVIFrameCount) As Boolean

#### Description

Used to retrieve the duration of the AVI file specified.

**Duration:** returns the duration of the AVI file, expressed in 100 nanoseconds units, returns 0 upon failure.

**FrameCount:** returns the total frame count if available, otherwise returns 0.

VideoWidth: width of the video

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Experience the Power and Ease of Use of HelpNDoc for CHM Help File Generation](#)

## AVIHeaderInfo

### TVideoGrabber.AVIHeaderInfo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the header information about the video clip specified.

#### Declaration

**function** AVIHeaderInfo (AVIFile: **string**; HeaderAttribute: THeaderAttribute): **string**;

wchar\_t \* \_\_fastcall AVIHeaderInfo(wchar\_t \*AVIFile, THeaderAttribute HeaderAttribute);

Function AVIHeaderInfo (AVIFile As String, HeaderAttribute as THeaderAttribute) as String

#### Description

Used to retrieve the [THeaderAttribute](#) information about the video clip specified.

When the clip is opened a first time, all the whole [THeaderAttribute](#) attributes are read, so the clip is not re-opened during the subsequent calls, unless a different file name is specified.

E.g.:

```
procedure TfrmMainForm.Button2Click(Sender: TObject);
begin
    Showmessage (VideoGrabber.AVIHeaderInfo('MyFirstVideoClip.avi', ha_Title));
    Showmessage (VideoGrabber.AVIHeaderInfo('MySecondVideoClip.asf', ha_Copyright));
end;
```

#### See Also

[Player features](#) [Recording methods and properties](#) [TAVIMuxConfig](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnPlayerBufferingData](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioChannelRenderMode](#) [AudioRecording](#) [AudioStreamNumber](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [Encoder](#) [SetInt](#) [FastForwardPlayer](#) [HoldRecording](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [IsRecordingPaused](#) [Last Clip Played](#) [Last Recording](#) [FileName](#) [MP4NeedsReindexing](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PauseRecording](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [ShowDialog](#) [SourceStream](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopPlayer](#) [StopRecording](#) [StoragePath](#) [SynchronizationRole](#) [Synchronized](#) [SyncPreview](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

#### AVIInfo

### TVideoGrabber.AVIInfo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns information about the specified video clip.

#### Declaration

**function** AVIInfo (AVIFile: **string**; **out** Duration: LargeInteger; **out** FrameCount: LargeInteger; **out** VideoWidth: LongInt; **out** VideoHeight: LongInt; **out** VideoFrameRateFps: Double; **out** AvgBitRate: LongInt; **out** AudioChannels: LongInt; **out** AudioSamplesPerSec: LongInt; **out** AudioBitsPerSample: LongInt; **out** VideoCodec: string; **out** AudioCodec: string): Boolean;

**bool** \_\_fastcall AVIInfo (wchar\_t \*AVIFile, \_\_int64 &Duration, \_\_int64 &FrameCount, **int** &VideoWidth, **int** &VideoHeight, **double** &VideoFrameRateFps, **int** &AvgBitRate, **int** &AudioChannels, **int** &AudioSamplesPerSec, **int** &AudioBitsPerSample, wchar\_t \*&VideoCodec, wchar\_t \*&AudioCodec)

Function AVIInfo(AVIFile As String, Duration, FrameCount, VideoWidth, VideoHeight, VideoFrameRateFps, AvgBitRate, AudioChannels, AudioSamplesPerSec, AudioBitsPerSample, VideoCodec, AudioCodec) As Boolean

#### Description

Used to retrieve information about the video clip specified as parameter.

**Duration:** returns the duration of the AVI file, expressed in 100 nanoseconds units, returns 0 upon failure,

**FrameCount:** returns the total frame count if available, otherwise returns 0,

**VideoWidth:** returns the width of the video

**VideoHeight:** returns the height of the video

**VideoFrameRateFps:** returns the frame rate in frames per second

**AvgBitRate:** returns the average bit rate of the clip

**AudioChannels:** returns the number of audio channels

**AudioSamplesPerSec:** returns the audio sample frequency in Hz

**AudioBitsPerSample:** returns the number of bits per sample

**VideoCodec:/cf0** returns the name of the video codec in which the video stream of the clip is encoded (if any)

**AudioCodec:/cf0** returns the name of the audio codec in which the audio stream of the clip is encoded (if any)

E.g.:

```
procedure TForm1.Button1Click(Sender: TObject);
var
    Duration: int64;
    FrameCount: int64;
    VideoWidth: LongInt;
    VideoHeight: LongInt;
    VideoCodec: string;
    AudioCodec: string;
    VideoFrameRateFps: Double;
    AvgBitRate: LongInt;
    AudioChannels: LongInt;
    AudioSamplesPerSec: LongInt;
    AudioBitsPerSample: LongInt;
begin
    if VideoGrabber.AVIInfo (edtPlayerClip.Text, Duration, FrameCount, VideoWidth, VideoHeight, VideoFrameRateFps, AvgBitRate, AudioChannels, AudioSamplesPerSec, AudioBitsPerSample, VideoCodec, AudioCodec) then
        ShowMessage('');
        ShowMessage(edtPlayerClip.Text + ':');
        ShowMessage('duration (in sec): ' + FormatFloat ('0.00', Duration / 1000000));
        ShowMessage('frame count: ' + IntToStr (FrameCount));
        ShowMessage('video width: ' + IntToStr (VideoWidth));
        ShowMessage('video height: ' + IntToStr (VideoHeight));
        ShowMessage('video frame rate (fps): ' + FormatFloat ('0.00', VideoFrameRateFps));
        ShowMessage('average bit rate (kb/s): ' + IntToStr (AvgBitRate div 1024));
        ShowMessage('audio channels: ' + IntToStr (AudioChannels));
        ShowMessage('audio samples/sec: ' + IntToStr (AudioSamplesPerSec));
```

```

        ShowMessage('audio bits per sample: ' + IntToStr (AudioBitsPerSample));
        ShowMessage('video codec: ' + VideoCodec);
        ShowMessage('audio codec: ' + AudioCodec);
    end
    else begin
        ShowMessage(edtPlayerClip.Text + ' clip not found.');
```

## See Also

[Player features](#) [Recording methods and properties](#) [TAVIMuxConfig](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnPlayerBufferingData](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioChannelRenderMode](#) [AudioRecording](#) [AudioStreamNumber](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo2](#) [ClosePlayer](#) [Encoder](#) [SetInt](#) [FastForwardPlayer](#) [HoldRecording](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [IsRecordingPaused](#) [Last Clip Played](#) [Last Recording](#) [FileName](#) [MP4NeedsReindexing](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PauseRecording](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [ShowDialog](#) [SourceStream](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopPlayer](#) [StopRecording](#) [StoragePath](#) [SynchronizationRole](#) [Synchronized](#) [SyncPreview](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## AVIInfo2

### TVideoGrabber.AVIInfo2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to retrieve information about the specified video clip.

#### Declaration

**function** AVIInfo2 (AVIFile: **string**; AVIInfoType: TAVIInfoType): **string**;

wchar\_t \* \_\_fastcall AVIInfo2(wchar\_t \*AVIFile, TAVIInfoType AVIInfoType);

Function AVIInfo2 (AVIFile as string; AVIInfoType as TAVIInfoType) as String

#### Description

This function returns the [TAVIInfoType](#) value as a string for the video clip specified as parameter.  
E.g.:

```

Dim VideoCodec as string
VideoCodec = AVIInfo2 ("myvideoclip.avi", av_VideoCodec)
```

Dim ClipDuration as string  
ClipDuration = AVIInfo2 ("myvideoclip.avi", av\_Duration)

Note: if the clip file name does not change, the clip is opened only one time. All the values are read and cached.

Therefore all the subsequent calls to AVIInfo2 with the same file name do not reopen the video clip.

The possible values returned by AVIInfo2 depending on TAVIInfoType are (av\_Duration, av\_FrameCount, av\_VideoWidth, av\_VideoHeight, av\_VideoFrameRateFps, av\_VideoCodec, av\_AudioCodec, av\_AvgBitRate, av\_AudioChannels, av\_AudioSamplesPerSec, av\_AudioBitsPerSample);

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last Clip Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## CameraControlAuto

### TVideoGrabber.CameraControlAuto

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Reports if a [TCameraControl](#) setting is in "auto" mode.

#### Declaration

**function** CameraControlAuto(Setting: TCameraControl): Boolean;

**bool** \_\_fastcall CameraControlAuto(TCameraControl Setting)

Function CameraControlAuto(Setting As TxCameraControl) As Boolean

#### Description

Used to know if a [TCameraControl](#) setting is in "auto" mode or "manual" mode.  
Returns "true" if the setting is in "auto" mode, "false" if it is in "manual" mode.

#### See Also

[TCameraControl](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## CameraControlDefault

### TVideoGrabber.CameraControlDefault

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)



Sets a camera control setting to its default value.

#### Declaration

**function** CameraControlDefault(Setting: TCameraControl): LongInt;

**int** \_\_fastcall CameraControlDefault(TCameraControl Setting)

Function CameraControlDefault(Setting As TxCameraControl) As Long

#### Description

Used to set a camera control setting to its default value. Returns true upon success.

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation with HelpNDoc's Project Analyzer Features](#)

## CameraControlMax

### TVideoGrabber.CameraControlMax

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Gets the maximum value of a given [TCameraControl](#) setting.

#### Declaration

**function** CameraControlMax(Setting: TCameraControl): LongInt;

**int** \_\_fastcall CameraControlMax(TCameraControl Setting)

Function CameraControlMax(Setting As TxCameraControl) As Long

#### Description

Retrieves the maximum value for a given [TCameraControl](#) setting, if available for the current video capture device. Returns MAXINT upon failure.

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

## CameraControlMin

### TVideoGrabber.CameraControlMin

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Gets the minimum value of a given [TCameraControl](#) setting.

#### Declaration

**function** CameraControlMin(Setting: TCameraControl): LongInt;

**int** \_\_fastcall CameraControlMin(TCameraControl Setting)



Function CameraControlMin(Setting As TxCameraControl) As Long

### Description

Retrieves the minimum value for a given [TCameraControl](#) setting, if available for the current video capture device. Returns MAXINT upon failure.

### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

## CameraControlStep

### TVideoGrabber.CameraControlStep

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Gets the stepping value of a given [TCameraControl](#) setting.

### Declaration

**function** CameraControlStep(Setting: TCameraControl): LongInt;

**int** \_\_fastcall CameraControlStep(TCameraControl Setting)

Function CameraControlStep(Setting As TxCameraControl) As Long

### Description

Retrieves the stepping value for a given [TCameraControl](#) setting, if available for the current video capture device. Returns MAXINT upon failure.

### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## CameraControlValue

### TVideoGrabber.CameraControlValue

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves a [TCameraControl](#) current value.

### Declaration

**function** GetCameraControl(Setting: TCameraControl): LongInt;

**int** \_\_fastcall GetCameraControl(TCameraControl Setting)

Function GetCameraControl(Setting As TxCameraControl) As Long

### Description

Used to retrieve the current value of a [TCameraControl](#) setting. Returns MAXINT upon failure.

### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#)

[CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## Cancel

### TVideoGrabber.Cancel

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Cancels any task currently running.

#### Declaration

**function** Cancel: Boolean;

**void** \_\_fastcall Cancel(**void**)

Function Cancel() As Boolean

#### Description

Cancels any task currently running (preview, recording, playback as well as reencoding).

#### Recording

If the recording is running, it is immediately stopped:

- if the recording had to be recompressed after capture ([CompressionMode](#) = cm\_CompressAfterRecording), this reencoding operation is cancelled,
- if the recording had to be recopied after capture ([PreallocCapFileEnabled](#) = true), this copy operation is cancelled.

#### Reencoding

The current reencoding operation is stopped immediately.

Returns true upon success.

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

## CanProcessMessages

### TVideoGrabber.CanProcessMessages

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to know if message-based processings can be applied

#### Declaration

**function** CanProcessMessages: Boolean;

**void** \_\_fastcall CanProcessMessages(**void**)

Function CanProcessMessages() As Boolean

#### Description

Used to know if message-based processings can be applied, mainly during the TVideoGrabber events like the [OnFrameCaptureCompleted](#) or [OnFrameOverlayUsing](#)... events.

This function returns false if the player is running and [UseClock](#) is disabled. In this case you must not invoke message-based code (e.g. like creating and displaying a form) during the [OnFrameCaptureCompleted](#) , [OnFrameProgress](#) or [OnFrameOverlayUsingDC](#) and other [OnFrameOverlayUsing](#)... events.

## CaptureFrameRenderedTo

### TVideoGrabber.CaptureFrameRenderedTo

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Captures the frame being rendered

#### Declaration

**function** CaptureFrameRenderedTo (DisplayIndex: LongInt; Dest: TFrameCaptureDest; FileName: **string**): Boolean;

**bool** \_\_fastcall CaptureFrameRenderedTo(int DisplayIndex, TFrameCaptureDest Dest, System::wchar\_t \*FileName);

#### Description

Capture the current frame being displayed in the video renderer.  
Works like [CaptureFrameSyncTo](#), but does not require the frame grabber to be enabled, allowing to capture frames and saving CPU.  
This function is supported only by the following [video renderers](#):  
vr\_EVR, vr\_VMR9, vr\_VMR7, vr\_StandardRenderer, vr\_madVR

Note :

Invoking this function with vr\_StandardRenderer produces a (very short) pause/resume of the video stream.

It is instant with other video renderers.

#### See Also

[Frame capture features](#)

## CaptureFrameSyncTo

### TVideoGrabber.CaptureFrameSyncTo

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Invokes [CaptureFrameTo](#) and waits until completion.

#### Declaration

**function** CaptureFrameSyncTo(Dest: TFrameCaptureDest; FileName: **String**): Boolean;

**bool** \_\_fastcall CaptureFrameSyncTo(TFrameCaptureDest Dest, wchar\_t \*FileName)

Function CaptureFrameSyncTo(Dest As TFrameCaptureDest, FileName As String) As Boolean

#### Description

Invokes [CaptureFrameTo](#) and waits until the frame to be captured, or the graph stopped.  
Returns true upon success, false if the capture failed.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#)

[JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#)  
[Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#)  
[WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## CaptureFrameTo

### TVideoGrabber.CaptureFrameTo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Captures asynchronously the next video frame to memory or to files automatically.

#### Declaration

**function** CaptureFrameTo(Dest: TFrameCaptureDest; FileName: **String**): Boolean;

**bool** \_\_fastcall CaptureFrameTo(TFrameCaptureDest Dest, wchar\_t \*FileName)

Function CaptureFrameTo(Dest As TFrameCaptureDest, FileName As String) As Boolean

#### Description

Captures asynchronously the next video frame, depending of the Dest parameter ([TFrameCaptureDest](#) type):

- **fc\_TBitmap**: to a memory Bitmap,
- **fc\_BmpFile**: to a BMP file
- **fc\_JpegFile**: to a JPEG file,
- **fc\_Clipboard**: to the clipboard (CF\_BITMAP format)

The [frame grabber](#) must be enabled to use this function.

The video frame is returned asynchronously by the [OnFrameCaptureCompleted](#) event.

To let TVideoGrabber generate automatically the file name when capturing a BMP or JPEG image to a file (Dest = fc\_BmpFile or Dest = fc\_JpegFile) , **pass an empty string in the FileName property**.

In this case a file name will be generated automatically in the [StoragePath](#) directory according to [AutoFileName](#) and [AutoFilePrefix](#) .

E.g.:

```
CaptureFrameTo (fc_JpegFile, '');
```

Otherwise you can specify yourself in the FileName property the path/file name to save the BMP or JPEG image.

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#)  
[BurstType](#) [CaptureFrameSyncTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#)  
[FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#)  
[GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#)  
[JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#)  
[Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#)  
[WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## ClearHeaderAttributes

### TVideoGrabber.ClearHeaderAttributes

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Resets the AVI and ASF header attributes.

#### Declaration

**procedure** ClearHeaderAttributes;

**void** \_\_fastcall ClearHeaderAttributes(**void**)

Public Overridable Sub ClearHeaderAttributes()

#### Description

Used to Reset all the AVI and ASF header attributes that have been set by invoking [SetHeaderAttribute](#).

#### See Also

[THeaderAttribute](#) [SetHeaderAttribute](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

## ClosePlayer

### TVideoGrabber.ClosePlayer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Closes the current video clip.

#### Declaration

**procedure** ClosePlayer;

**void** \_\_fastcall ClosePlayer(**void**)

Public Overridable Sub ClosePlayer()

#### Description

Closes the [PlayerFileName](#) video clip, previously opened with [OpenPlayer](#) .

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [HelpNDoc's Project Analyzer: Incredible documentation assistant](#)

## Create

### TVideoGrabber.Create

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Constructor.

## Declaration

**constructor** Create(Aowner: TComponent); **override**;

## Description

Constructor.

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## CreatePreallocCapFile

### TVideoGrabber.CreatePreallocCapFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Creates an huge preallocated file.

## Declaration

**function** CreatePreallocCapFile: Boolean;

**bool** \_\_fastcall CreatePreallocCapFile(**void**)

Function CreatePreallocCapFile() As Boolean

## Description

Used to create an huge preallocated file, according to [PreallocCapFileFileName](#) and [PreallocCapFileSizeInMB](#).

## See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileProgress](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## Decrypt\_File

### TVideoGrabber.Decrypt\_File

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Decrypts a video clip

## Declaration

**function** Decrypt\_File (InputFile: **string**; OutputFile: **string**; OverwriteIfExists: Boolean): Boolean;

**bool** \_\_fastcall Decrypt\_File(System::UnicodeString InputFile, System::UnicodeString OutputFile, **bool** OverwriteIfExists);

## Description

Note: if the input file exists but is not encrypted the function fails

## See Also

[TEncryptionMethod](#) [Encrypt\\_File](#) [EncryptionMethod](#) [SetDecryptionKey](#) [SetEncryptionKey](#)

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## Destroy

### TVideoGrabber.Destroy

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

[ber](#)

Destructor.

#### Declaration

**destructor** Destroy; **override**;

#### Description

Destructor.

Created with the Standard Edition of HelpNDoc: [Quickly and Easily Convert Your Word Document to an ePub or Kindle eBook](#)

## Display\_SetLocation

### TVideoGrabber.Display\_SetLocation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to set the location of the video window, when it is not embedded.

#### Declaration

**procedure** Display\_SetLocation(WindowLeft, WindowTop, WindowWidth, WindowHeight: LongInt);

**void** \_\_fastcall Display\_SetLocation(int WindowLeft, int WindowTop, int WindowWidth, int WindowHeight)

Public Overridable Sub Display\_SetLocation(ByVal WindowLeft As Integer, ByVal WindowTop As Integer, ByVal WindowWidth As Integer, ByVal WindowHeight As Integer)

#### Description

Use this method to set the video window location and size at a time (left, top, width, height), when the video window is not embedded in the TVideoGrabber component ([Display\\_Embedded](#) = false).

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## DrawBitmapOverFrame

### TVideoGrabber.DrawBitmapOverFrame

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Draws the bitmap passed as parameter over video frames.

#### Declaration



**function** DrawBitmapOverFrame (Bitmap: Graphics.TBitmap; LeftLocation: LongInt; TopLocation: LongInt; bmWidth: LongInt; bmHeight: LongInt; Transparent: Boolean; UseTransparentColor: Boolean; TransparentColorValue: TColor; AlphaBlend: Boolean; AlphaBlendValue: LongInt): Boolean;

**bool** \_\_fastcall DrawBitmapOverFrame(Graphics::TBitmap \*Bitmap, **int** LeftLocation, **int** TopLocation, **int** bmWidth, **int** bmHeight, **bool** Transparent, **bool** UseTransparentColor, **TColor** TransparentColorValue, **bool** AlphaBlend, **int** AlphaBlendValue)

Function DrawBitmapOverFrame(BitmapHandle As Long, LeftLocation As Long, TopLocation As Long, bmWidth As Long, bmHeight As Long, Transparent As Boolean, UseTransparentColor As Boolean, TransparentColorValue As Long, AlphaBlend As Boolean, AlphaBlendValue As Long) As Boolean

### Description

Used to draw bitmaps over video frames.

### Location

The bitmap is drawn at the LeftLocation, TopLocation position over the video frame (expressed in pixels).

### Size

If the **bmWidth** or **bmHeight** parameter is different from the bitmap's width or height, the bitmap is stretched to **bmWidth** and **bmHeight**.

### Transparency

- if the **Transparent** parameter is true, the image background is not drawn.
- if the **Transparent** parameter is true and the **UseTransparentColor** parameter is true, the **TransparentColorValue** value is used for transparency.

### Alpha blending

If the **AlphaBlend** parameter is true, a progressive bitmap transparency is applied, depending of the **AlphaBlendValue**, in a 0..255 range.

It is not possible to perform alphablending when the Transparent parameter is true.

#### Important:

if the bitmap color format is not 32 bits, it will be converted into a 32 bit format. In other words, for better performance pass a bitmap already in 32 bits color format.

**The DrawBitmapOverFrame function can be called ONLY from the [OnFrameOverlayUsingDC](#) event.** See the MainDemo project for sample code.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)

[SetTextOverlay\\_GradientMode](#)
[SetTextOverlay\\_HighResFont](#)
[SetTextOverlay\\_Left](#)
[SetTextOverlay\\_Right](#)
[SetTextOverlay\\_Scrolling](#)
[SetTextOverlay\\_ScrollingSpeed](#)
[SetTextOverlay\\_Shadow](#)
[SetTextOverlay\\_ShadowColor](#)
[SetTextOverlay\\_ShadowDirection](#)
[SetTextOverlay\\_String](#)
[SetTextOverlay\\_TargetDisplay](#)
[SetTextOverlay\\_Top](#)
[SetTextOverlay\\_Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay\\_Align](#)
[TextOverlay\\_BkColor](#)
[TextOverlay\\_CreateCustomFont](#)
[TextOverlay\\_CreateCustomFont2](#)
[TextOverlay\\_Enabled](#)
[TextOverlay\\_Font](#)
[TextOverlay\\_FontColor](#)
[TextOverlay\\_Left](#)
[TextOverlay\\_Right](#)
[TextOverlay\\_Scrolling](#)
[TextOverlay\\_ScrollingSpeed](#)
[TextOverlay\\_Selector](#)
[TextOverlay\\_Shadow](#)
[TextOverlay\\_ShadowColor](#)
[TextOverlay\\_ShadowDirection](#)
[TextOverlay\\_String](#)
[TextOverlay\\_Top](#)
[TextOverlay\\_Transparent](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

## DualDisplay\_SetLocation

### TVideoGrabber.DualDisplay\_SetLocation

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Methods](#)

Same as [Display\\_SetLocation](#) , but about the 2nd video window.

#### Declaration

**procedure** DualDisplay\_SetLocation(WindowLeft, WindowTop, WindowWidth, WindowHeight: LongInt);

**void** \_\_fastcall DualDisplay\_SetLocation(int WindowLeft, int WindowTop, int WindowWidth, int WindowHeight)

Public Overridable Sub DualDisplay\_SetLocation(ByVal WindowLeft As Integer, ByVal WindowTop As Integer, ByVal WindowWidth As Integer, ByVal WindowHeight As Integer)

#### Description

Same as [Display\\_SetLocation](#) , but about the 2nd video window.

#### See Also

[Dual display](#)
[Display\\_AlphaBlendEnabled](#)
[Display\\_AlphaBlendValue](#)
[Display\\_TransparentColorEnabled](#)
[Display\\_TransparentColorValue](#)
[DualDisplay\\_Active](#)
[DualDisplay\\_AlphaBlendEnabled](#)
[DualDisplay\\_AlphaBlendValue](#)
[DualDisplay\\_AutoSize](#)
[DualDisplay\\_Embedded](#)
[DualDisplay\\_FullScreen](#)
[DualDisplay\\_Height](#)
[DualDisplay\\_Left](#)
[DualDisplay\\_Monitor](#)
[DualDisplay\\_MouseMovesWindow](#)
[DualDisplay\\_PanScanRatio](#)
[DualDisplay\\_StayOnTop](#)
[DualDisplay\\_Top](#)
[DualDisplay\\_TransparentColorEnabled](#)
[DualDisplay\\_TransparentColorValue](#)
[DualDisplay\\_VideoHeight](#)
[DualDisplay\\_VideoPortEnabled](#)
[DualDisplay\\_VideoWidth](#)
[DualDisplay\\_VideoWindowHandle](#)
[DualDisplay\\_Visible](#)
[DualDisplay\\_Width](#)
[Monitor\\_Primary\\_Index](#)
[MonitorBounds](#)
[MonitorsCount](#)
[SetWindowTransparency](#)
[VideoRendererExternal](#)
[VideoRendererExternalIndex](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## DVDInfo

### TVideoGrabber.DVDInfo

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Methods](#)

DVD Info

#### Declaration

**function** DVDInfo(DVDRootDirectory: **string**; DVDInfoType: TDVDInfoType; TitleNumber: LongInt): Double;

**double \_\_fastcall** DVDInfo(System::wchar\_t \*DVDRootDirectory, TDVDInfoType DVDInfoType, int TitleNumber)

Function DVDInfo (DVDRootDirectory as string, DVDInfoType as TDVDInfoType, TitleNumber as Long) as Double

### Description

This function lets you retrieve information about the DVD.

**DVDRootDirectory:** path to the VIDEO\_TS folder

**DVDInfoType:** selects the [TDVDInfoType](#) type of information requested. Possible values:

dvd\_NumberOfVolumes  
 dvd\_TotalDuration  
 dvd\_NumberOfTitles  
 dvd\_TitleDuration  
 dvd\_TitleFrameRate  
 dvd\_SourceResolutionX  
 dvd\_SourceResolutionY  
 dvd\_TitleFrameCount

**TitleNumber:** number of the title requested (in the 1..n range) or 0 for the global information (dvd\_TotalDuration and dvd\_NumberOfTitles)

(note that the DVD data is cached, therefore the DVD in read only one time, other calls to DVDInfo just return the cached values, unless you specify a different DVD folder)

(all the values are returned as double)  
 E.g. to get:

- *the total duration of the DVD:*

```
double TotalDuration = Videograbber.DvdInfo ("e:dvd_TotalDuration, 0)
```

- *the number of titles in the DVD:*

```
int NumberOfTitles = (int) Videograbber.DvdInfo ("e:dvd_NumberOfTitles, 0)
```

- *the duration of the 1st title:*

```
double TitleDuration = Videograbber.DvdInfo ("e:dvd_TitleDuration, 1)
```

- *the number of frames of the 1st title:*

```
int TitleFrameCount = (int) Videograbber.DvdInfo ("e:dvd_TitleFrameCount, 1)
```

- *the frame rate of the 1st title:*

```
int TitleFrameRate = (int) Videograbber.DvdInfo ("e:dvd_TitleFrameRate, 1)
```

- *the width of the video resolution of the 1st title:*

```
int TitleVideoWidth = (int) Videograbber.DvdInfo ("e:dvd_SourceResolutionX, 1)
```

- *the height of the video resolution of the 1st title:*

```
int TitleVideoHeight = (int) Videograbber.DvdInfo ("e:dvd_SourceResolutionY, 1)
```

### See Also

[DVDTitle](#) [OpenDVD](#) [PlayerFileName](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly create a professional-quality documentation website with HelpNDoc](#)

---

## EnableMultiplexedInput

### TVideoGrabber.EnableMultiplexedInput

[Prev](#)

[Next](#)

[TVideoGrab](#) [Methods](#)

**ber**

Enables / disables a multiplexed input.

**Declaration**

**function** EnableMultiplexedInput(InputNumber: LongInt; Enable: Boolean): Boolean;

**bool** \_\_fastcall EnableMultiplexedInput(**int** InputNumber, **bool** Enable);

Function EnableMultiplexedInput(InputNumber as Long, Enable as Boolean) as Boolean

**Description**

Used to enable / disable a multiplexed input.

**See Also**

[TMultiplexedRole](#) [AssociateMultiplexedSlave](#) [MultiplexedInputEmulation](#) [MultiplexedRole](#)  
[MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

**EnableThreadMode****TVideoGrabber.EnableThreadMode**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)  
**ber**

Enables the thread mode

**Declaration**

procedure EnableThreadMode;

void EnableThreadMode()

Sub EnableThreadMode()

**Description**

To enable the thread mode, invoke EnableThreadMode just after creating the component.

**See Also**

[Opening a clip or an IP URL from a background thread without blocking the main thread](#) [Player features](#)  
[TOnThreadSync](#) [TThreadSyncPoint](#) [OnThreadSync](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

**Encoder\_CloseOutputFile****TVideoGrabber.Encoder\_CloseOutputFile**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)  
**ber**

Encoder\_NewOutputFile

**Declaration**

**function** Encoder\_CloseOutputFile (EncoderUniqueID: LongInt): Boolean;

**int** Encoder\_CloseOutputFile(**int** EncoderUniqueID);

**Description**

Close the file curently being written

## Encoder\_GetInt

### TVideoGrabber.Encoder\_GetInt

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Encoder\_GetInt

#### Declaration

**function** Encoder\_GetInt (EncoderID: LongInt; Setting: TEncoder\_int; **var** Value: LongInt): Boolean;**bool** Encoder\_GetInt(int EncoderID, TEncoder\_int Setting, **int** \*Value);

#### Description

Retrives an integer value from the [Datastead Encoder](#)

Useful mainly to get the total Kb written by invoking Encoder\_GetInt (ID, Enc\_Bytes\_Written\_kb\_readonly, Value)

#### See Also

[Datastead Encoder Pause/resume during recording](#) [Recording methods and properties](#) [Encoder Pause](#) [Encoder Resume](#) [Encoder SetInt](#) [Encoder SetStr](#) [Encoders CreateInstanceForRecording](#) [Encoders CreateInstanceForStreaming](#) [Encoders RemoveAllInstances](#) [Encoders RemoveInstance](#)

## Encoder\_NewOutputFile

### TVideoGrabber.Encoder\_NewOutputFile

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Encoder\_NewOutputFile

#### Declaration

**function** Encoder\_NewOutputFile (EncoderUniqueID: LongInt; OutputFile: **string**; OpenPaused: Boolean): Boolean;**int** Encoder\_NewOutputFile(int EncoderUniqueID, wchar\_t \*OutputFile, **bool** OpenPaused);

#### Description

Close the file curently being written, and starts recording to the new file name specified.

If OpenPaused is true, the file will not be written until [Encoder Resume](#) is invoked.

#### See Also

[Datastead Encoder Pause/resume during recording](#) [Recording methods and properties](#) [Encoder GetInt](#) [Encoder Pause](#) [Encoder Resume](#) [Encoder SetInt](#) [Encoder SetStr](#) [Encoders CreateInstanceForRecording](#) [Encoders CreateInstanceForStreaming](#) [Encoders RemoveAllInstances](#) [Encoders RemoveInstance](#)

## Encoder\_Pause

### TVideoGrabber.Encoder\_Pause

[Prev](#)[Next](#)

## [TVideoGrabber](#) [Methods](#)

Encoder\_Pause

### Declaration

**function** Encoder\_Pause (EncoderUniqueID: LongInt): Boolean;

**bool** Encoder\_Pause(int EncoderUniqueID);

### Description

Pauses the encoder:

- if recording to a file, stops writing to the file
- if streaming, stops streaming

The video preview does not stop.

### See Also

[Datastead Encoder Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveAllInstances](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## Encoder\_Resume

### TVideoGrabber.Encoder\_Resume

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

Encoder\_Resume

### Declaration

**function** Encoder\_Resume (EncoderUniqueID: LongInt): Boolean;

**bool** Encoder\_Resume(int EncoderUniqueID);

### Description

Resumes the encoder:

- if recording to a file, restart writing to the file
- if streaming, resume streaming

The video preview goes on without stop/restart

### See Also

[Datastead Encoder Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveAllInstances](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's HTML5 template](#)

## Encoder\_SetInt

### TVideoGrabber.Encoder\_SetInt

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

Encoder\_SetInt

### Declaration

**function** Encoder\_SetInt (EncoderID: LongInt; Setting: TEncoder\_int; Value: LongInt): Boolean;

**bool** Encoder\_SetInt(int EncoderID, TEncoder\_int Setting, int Value);

### Description

Sets an integer parameter of the [Datastead Encoder](#)

### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [GetInt](#) [Encoder](#) [Pause](#) [Encoder](#) [Resume](#) [Encoder](#) [SetStr](#) [Encoders](#) [CreateInstanceForRecording](#) [Encoders](#) [CreateInstanceForStreaming](#) [Encoders](#) [RemoveAllInstances](#) [Encoders](#) [RemoveInstance](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Elevate your documentation to new heights with HelpNDoc's built-in SEO

## Encoder\_SetStr

### TVideoGrabber.Encoder\_SetStr

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Encoder\_SetStr

### Declaration

**function** Encoder\_SetStr (EncoderID: LongInt; Setting: TEncoder\_str; Value: **string**): Boolean;

**bool** Encoder\_SetStr(int EncoderID, TEncoder\_str Setting, wchar\_t \*Value);

### Description

Sets a string parameter of the [Datastead Encoder](#)

### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [Recording methods and properties](#) [Encoder](#) [GetInt](#) [Encoder](#) [Pause](#) [Encoder](#) [Resume](#) [Encoder](#) [SetInt](#) [Encoders](#) [CreateInstanceForRecording](#) [Encoders](#) [CreateInstanceForStreaming](#) [Encoders](#) [RemoveAllInstances](#) [Encoders](#) [RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: Experience a User-Friendly Interface with HelpNDoc's Documentation Tool

## Encoders\_CreateInstanceForRecording

### TVideoGrabber.Encoders\_CreateInstanceForRecording

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Encoders\_CreateInstanceForRecording

### Declaration



**function** Encoders\_CreateInstanceForRecording (OutputFile: **string**): LongInt;

**int** Encoders\_CreateInstanceForRecording(**wchar\_t** \* OutputFile);

#### Description

Adds a [Datastead Encoder](#) instance to the graph, to write to a new file.

Returns a unique ID to this encoder.

This unique ID will be required to invoke [Encoder\\_SetInt](#), [Encoder\\_GetInt](#), [Encoder\\_SetStr](#), [Encoder\\_Pause](#), [Encoder\\_Resume](#), [Encoder\\_RecordToNewFileNow](#), or [Encoders\\_RemoveInstance](#).

#### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveAllInstances](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

## Encoders\_CreateInstanceForStreaming

### TVideoGrabber.Encoders\_CreateInstanceForStreaming

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Encoders\_CreateInstanceForStreaming

#### Declaration

**function** Encoders\_CreateInstanceForStreaming (OutputURL: **string**): LongInt;

**int** Encoders\_CreateInstanceForStreaming(**wchar\_t** \*OutputURL);

#### Description

Adds a [Datastead Encoder](#) instance to the graph, to stream to a new destination.

Returns a unique ID to this encoder.

This unique ID will be required to invoke [Encoder\\_SetInt](#), [Encoder\\_GetInt](#), [Encoder\\_SetStr](#), [Encoder\\_Pause](#), [Encoder\\_Resume](#), [Encoder\\_RecordToNewFileNow](#), or [Encoders\\_RemoveInstance](#).

#### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_RemoveAllInstances](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## Encoders\_RemoveAllInstances

### TVideoGrabber.Encoders\_RemoveAllInstances

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Encoders\_RemoveAllInstances

#### Declaration

**procedure** Encoders\_RemoveAllInstances();

**void** Encoders\_RemoveAllInstances(**void**);

#### Description

Remove all the [Datastead Encoder](#) instances previously added by [Encoders\\_CreateInstanceForRecording](#) or [Encoders\\_CreateInstanceForStreaming](#).

#### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveInstance](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Edit and Export Markdown Documents](#)

## Encoders\_RemoveInstance

### TVideoGrabber.Encoders\_RemoveInstance

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Encoders\_RemoveInstance

#### Declaration

**function** Encoders\_RemoveInstance (EncoderUniqueID: LongInt): Boolean;

**bool** Encoders\_RemoveInstance(**int** EncoderUniqueID);

#### Description

Remove all the [Datastead Encoder](#) instances previously added by [Encoders\\_CreateInstanceForRecording](#) or [Encoders\\_CreateInstanceForStreaming](#).

#### See Also

[Datastead Encoder](#) [Pause/resume during recording](#) [Recording methods and properties](#) [Encoder\\_GetInt](#) [Encoder\\_Pause](#) [Encoder\\_Resume](#) [Encoder\\_SetInt](#) [Encoder\\_SetStr](#) [Encoders\\_CreateInstanceForRecording](#) [Encoders\\_CreateInstanceForStreaming](#) [Encoders\\_RemoveAllInstances](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your PDF Protection with These Simple Steps](#)

## Encrypt\_File

### TVideoGrabber.Encrypt\_File

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Encrypts a video clip

#### Declaration

**function** Encrypt\_File (InputFile: **string**; OutputFile: **string**; OverwriteIfExists: Boolean): Boolean;

**bool** \_\_fastcall Encrypt\_File(System::UnicodeString InputFile, System::UnicodeString OutputFile, **bool** OverwriteIfExists);

#### Description

Note: if the input file exists but is already encrypted the function fails

#### See Also

[TEncryptionMethod](#) [Decrypt\\_File](#) [EncryptionMethod](#) [SetDecryptionKey](#) [SetEncryptionKey](#)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## EnumerateWindows

### TVideoGrabber.EnumerateWindows

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used in screen recording to retrieve information about visible windows.

#### Declaration

**function** EnumerateWindows: Boolean;

**bool** \_\_fastcall EnumerateWindows(**void**);

Function EnumerateWindows as Boolean

#### Description

Helper function that can be used for the screen recording of a window.

Immediately after Invoking EnumerateWindows, information about all the visible windows will be returned by the [OnEnumerateWindows](#) event, that will occur for each visible window.

See the "[Recording a window](#)" chapter for more information.

#### See Also

[TOnEnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#)  
[ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#)  
[SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

---

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

---

## Facebook\_GoLive\_ReturnStreamURL

### TVideoGrabber.Facebook\_GoLive\_ReturnStreamURL

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the streaming URL of a Facebook live session

#### Declaration

**function** Facebook\_GoLive\_ReturnStreamURL (EndPointURL: **string**; Title: **string**; Description: **string**; Status: **string**; AccessToken: **string**; caBundlePath: **string**): **string**;

System::UnicodeString \_\_fastcall Facebook\_GoLive\_ReturnStreamURL(System::UnicodeString EndPointURL, System::UnicodeString Title, System::UnicodeString Description, System::UnicodeString Status, System::UnicodeString AccessToken, System::UnicodeString caBundlePath);

#### Description

Used to obtain the RTMP URL of a Facebook endpoint, in order to pass it to the Multipurpose Encoder for Facebook live streaming.

#### C# sample code:

```
string strEndPointURL, strAccessToken, strRTMPURL;
strEndPointURL =
"https://graph.facebook.com/v23.0/64572161632085/live_videos";
strAccessToken =
"EAA LucaCaoyABPayE9rl4agr5hZB6YjwVxUS9bVg2hCcR089kiSt0lA10smuVFxZAZAyspzW7Uh6M
lWZCocsv7pYy2AE4tZCnLK4iVcAcYkjOHHWNUD5omifalyMeL8ZBA0qm53TlSnDmGjVTtlpiZBMyx3
ggHZBKkuuZCDs9TNN9IlUp8z5nWAoNFsZCJq8pH8wvBGUDfFOV4y18ZBS4C5CZC7lUrhlaogWB8XS1
xAtZC3WDJ2t4efsuurGeZCRQo3tLYZD";
// Assuming you have a method similar to the helphi function:
strRTMPURL = VideoGrabber.Facebook_GoLive_ReturnStreamURL(
strEndPointURL,
"fb streaming test",
"my description",
"LIVE_NOW",
strAccessToken,
"
```

);

**Delphi sample code:**

```

procedure TestFacebookGoLive;
var strEndPointURL, strAccessToken, strRTMPURL: string;
begin
  strEndPointURL :=
    'https://graph.facebook.com/v23.0/64572161632085/live_videos';
  strAccessToken :=
    'EAALucaCaoyABPayE9rl4agr5hZB6YjwVxUS9bVg2hCcR089kiSt01A10smuVFxZAZAyspzW7Uh6M
    lWZCocsv7pYy2AE4tZCnLK4iVcAcYkjOHWNUD5omifalyMeL8ZBA0qm53TlsnDmGjVTtlpiZBMyx3
    ggHZBKkuuZCDs9TNN9IlUp8z5nWAoNFsZCJq8pH8wvBGUDfFOV4y18ZBS4C5CZC7lUrhlaogWB8XS1
    xAtZC3WDJ2t4efsuurGeZCRQo3tLYZD';
  strRTMPURL := VideoGrabber.Facebook_GoLive_ReturnStreamURL(strEndPointURL, 'fb
    streaming test', 'my description', 'LIVE_NOW', strAccessToken, '');
  if copy(strRTMPURL, 1, 5) <> 'ERROR' then begin
    showmessage(strRTMPURL);
  end;
end;
end;

```

---

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

---

**FastForwardPlayer****TVideoGrabber.FastForwardPlayer**[Prev](#)[Next](#)
[TVideoGrabber](#)   [Methods](#)

Plays a video clip forwards at a different speed.

**Declaration**

**procedure** FastForwardPlayer;

**void** \_\_fastcall FastForwardPlayer(**void**)

Public Overridable Sub FastForwardPlayer()

**Description**

Used to play the clip forwards faster or slower than the normal speed.

The fast forward speed is the normal speed x [PlayerFastSeekRatio](#) .

**Restriction:** this feature is available only with seekable (indexed) AVI clips.

**See Also**

[Player features](#)
[TOnPlayerStateChanged](#)
[TPlayerState](#)
[TOnPlayerBufferingData](#)
[AudioChannelRenderMode](#)
[AudioStreamNumber](#)
[AutoStartPlayer](#)
[AVIDuration](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[AVIInfo2](#)
[ClosePlayer](#)
[IsPlayerAudioStreamAvailable](#)
[IsPlayerVideoStreamAvailable](#)
[Last\\_Clip\\_Played](#)
[MP4NeedsReindexing](#)
[OnPlayerBufferingData](#)
[OnPlayerEndOfStream](#)
[OnPlayerStateChanged](#)
[OnPlayerUpdateTrackbarPosition](#)
[OpenPlayer](#)
[OpenPlayerAtFramePositions](#)
[OpenPlayerAtTimePositions](#)
[PausePlayer](#)
[PlayerAudioCodec](#)
[PlayerAudioRendering](#)
[PlayerDuration](#)
[PlayerDVSize](#)
[PlayerFastSeekSpeedRatio](#)
[PlayerFileName](#)
[PlayerForcedCodec](#)
[PlayerFrameCount](#)
[PlayerFramePosition](#)
[PlayerFrameRate](#)
[PlayerFrameStep](#)
[PlayerRefreshPausedDisplay](#)
[PlayerRefreshPausedDisplayFrameRate](#)
[PlayerSpeedRatio](#)
[PlayerTimePosition](#)
[PlayerTrackBar](#)
[PlayerTrackBarScale](#)
[PlayerTrackBarSynchronise](#)
[PlayerVideoCodec](#)
[RewindPlayer](#)
[RunPlayer](#)
[RunPlayerBackwards](#)
[ShowDialog](#)
[SourceStream](#)
[StopPlayer](#)
[SynchronizationRole](#)
[Synchronized](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoPlayableWhileRecording](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

---

Created with the Standard Edition of HelpNDoc: [Easy EBook and documentation generator](#)

---

## FindIndexInListByName

### TVideoGrabber.FindIndexInListByName

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Search the index of an item in a list of strings, by name or substring

#### Declaration

**function** FindIndexInListByName(List: **String**; SearchedString: **String**; IsSubString: boolean; IgnoreCase: Boolean): LongInt;

**int** \_\_fastcall FindIndexInListByName(**UnicodeString** List, **UnicodeString** SearchedString, **bool** IsSubString, **bool** IgnoreCase)

Function FindIndexInListByName(List As String, SearchedString As String, IsSubString As Boolean, IgnoreCase As Boolean) As Long

#### Description

Used to find the index of an item in a list of strings, searching by the name of the item or a substring that identifies the item.

E.g., for a given video capture device, the [VideoInputs](#) list returns:

**Composite**  
**SVideo**  
**Tuner**

Therefore the index of "Composite" is 0, the index of "SVideo" is 1, and the index of "Tuner" is 2.

E.g. to programmatically select the "Composite" input. Proceed as follows:

```
procedure TfrmMainForm.Button1Click(Sender: TObject);
var
    i: LongInt;
begin
    i := VideoGrabber.FindIndexInListByName (VideoGrabber.VideoInputs, 'Composite', false, true);
    if i > -1 then begin // if this input exists...
        VideoGrabber.VideoInput := i;
    end;
end;
```

Or to select the same input by using only "Compos" as substring of the input name:

```
procedure TfrmMainForm.Button1Click(Sender: TObject);
var
    i: LongInt;
begin
    i := VideoGrabber.FindIndexInListByName (VideoGrabber.VideoInputs, 'Compos', true, true);
    if i > -1 then begin // if this input exists...
        VideoGrabber.VideoInput := i;
    end;
end;
```

#### See Also

[OnVideoDeviceSelected](#) [StoreDeviceSettingsInRegistry](#)

## GetAudioCodec

### TVideoGrabber.GetAudioCodec

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Current audio codec

#### Declaration

**function** GetAudioCodec: **string**;

System::wchar\_t \* \_\_fastcall GetAudioCodec()

Function GetAudioCodec as String

#### Description

Returns the current audio codec beeing used.

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## GetCameraExposure

### TVideoGrabber.GetCameraExposure

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to set the camera exposure as a double value, e.g. 1/8 s, 1/16 s, 1/32 s, etc...  
Retrieves the camera exposure as double value

#### Declaration

**function** GetCameraExposure: Double;

**double** \_\_fastcall GetCameraExposure(**void**);

Function GetCameraExposure() as Double

#### Description

Used to retrieve the camera exposure as a double value, e.g. 1/8 s, 1/16 s, 1/32 s, etc...

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## GetCameraExposureAsString

### TVideoGrabber.GetCameraExposureAsString

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the camera exposure value as a string

#### Declaration

**function** GetCameraExposureAsString: **string**;

wchar\_t \* \_\_fastcall GetCameraExposureAsString();

Function GetCameraExposureAsString as String

### Description

Used to retrieve the camera exposure as a fractional string, e.g. "1/8" s, "1/16" s, "1/32" s, etc...

### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## GetDisplayActive

### TVideoGrabber.GetDisplayActive

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayActive](#)

### Declaration

**function** GetDisplayActive (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayActive(int DisplayIndex);

function GetDisplayActive (DisplayIndex as Long) as **bool**

### Description

see [SetDisplayActive](#)

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## GetDisplayAlphaBlendEnabled

### TVideoGrabber.GetDisplayAlphaBlendEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayAlphaBlendEnabled](#)

### Declaration

**function** GetDisplayAlphaBlendEnabled (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayAlphaBlendEnabled(int DisplayIndex);

function GetDisplayAlphaBlendEnabled (DisplayIndex as Long) as **bool**

### Description

see [SetDisplayAlphaBlendEnabled](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## GetDisplayAlphaBlendValue

### TVideoGrabber.GetDisplayAlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)



see [SetDisplayAlphaBlendValue](#)

#### Declaration

**function** GetDisplayAlphaBlendValue (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayAlphaBlendValue(**int** DisplayIndex);

function GetDisplayAlphaBlendValue (DisplayIndex as Long) as **Long**

#### Description

see [SetDisplayAlphaBlendValue](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## GetDisplayAspectRatio

### TVideoGrabber.GetDisplayAspectRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayAspectRatio](#)

#### Declaration

**function** GetDisplayAspectRatio (DisplayIndex: LongInt): TAspectRatio;

TAspectRatio \_\_fastcall GetDisplayAspectRatio(**int** DisplayIndex);

function GetDisplayAspectRatio (DisplayIndex as Long) as TAspectRatio

#### Description

see [SetDisplayAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## GetDisplayAutoSize

### TVideoGrabber.GetDisplayAutoSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayAutoSize](#)

#### Declaration

**function** GetDisplayAutoSize (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayAutoSize(**int** DisplayIndex);

function GetDisplayAutoSize (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayAutoSize](#)

Created with the Standard Edition of HelpNDoc: [Elevate your documentation to new heights with HelpNDoc's built-in SEO](#)

## GetDisplayEmbedded

### TVideoGrabber.GetDisplayEmbedded

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayEmbedded](#)

#### Declaration

**function** GetDisplayEmbedded (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayEmbedded(**int** DisplayIndex);

function GetDisplayEmbedded (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayEmbedded](#)

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## GetDisplayFullScreen

### TVideoGrabber.GetDisplayFullScreen

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayFullScreen](#)

#### Declaration

**function** GetDisplayFullScreen (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayFullScreen(**int** DisplayIndex);

function GetDisplayFullScreen (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayFullScreen](#)

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## GetDisplayHeight

### TVideoGrabber.GetDisplayHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayHeight](#)

#### Declaration

**function** GetDisplayHeight (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayHeight(**int** DisplayIndex);

function GetDisplayHeight (DisplayIndex as Long) as Long

#### Description

see [SetDisplayHeight](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## GetDisplayLeft

### TVideoGrabber.GetDisplayLeft

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

see [SetDisplayLeft](#)

### Declaration

**function** GetDisplayLeft (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayLeft(int DisplayIndex);

function GetDisplayLeft (DisplayIndex as Long) as Long

### Description

see [SetDisplayLeft](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## GetDisplayMonitor

### TVideoGrabber.GetDisplayMonitor

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Methods](#)

see [SetDisplayMonitor](#)

### Declaration

**function** GetDisplayMonitor (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayMonitor(int DisplayIndex);

function GetDisplayMonitor (DisplayIndex as Long) as Long

### Description

see [SetDisplayMonitor](#)

Created with the Standard Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

## GetDisplayMouseMovesWindow

### TVideoGrabber.GetDisplayMouseMovesWindow

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Methods](#)

see [SetDisplayMouseMovesWindow](#)

### Declaration

**function** GetDisplayMouseMovesWindow (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayMouseMovesWindow(int DisplayIndex);

function GetDisplayMouseMovesWindow (DisplayIndex as Long) as **bool**

### Description

see [SetDisplayMouseMovesWindow](#)

Created with the Standard Edition of HelpNDoc: [Experience a User-Friendly Interface with HelpNDoc's Documentation Tool](#)

## GetDisplayPanScanRatio

### TVideoGrabber.GetDisplayPanScanRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayPanScanRatio](#)

#### Declaration

**function** GetDisplayPanScanRatio (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayPanScanRatio(**int** DisplayIndex);

function GetDisplayPanScanRatio (DisplayIndex as Long) as Long

#### Description

see [SetDisplayPanScanRatio](#)

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## GetDisplayStayOnTop

### TVideoGrabber.GetDisplayStayOnTop

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayStayOnTop](#)

#### Declaration

**function** GetDisplayStayOnTop (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayStayOnTop(**int** DisplayIndex);

function GetDisplayStayOnTop (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayStayOnTop](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## GetDisplayTop

### TVideoGrabber.GetDisplayTop

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayTop](#)

#### Declaration

**function** GetDisplayTop (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayTop(**int** DisplayIndex);

function GetDisplayTop (DisplayIndex as Long) as Long

#### Description

see [SetDisplayTop](#)

## GetDisplayTransparentColorEnabled

### TVideoGrabber.GetDisplayTransparentColorEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayTransparentColorEnabled](#)

#### Declaration

**function** GetDisplayTransparentColorEnabled (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayTransparentColorEnabled(**int** DisplayIndex);

function GetDisplayTransparentColorEnabled (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayTransparentColorEnabled](#)

## GetDisplayTransparentColorValue

### TVideoGrabber.GetDisplayTransparentColorValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayTransparentColorValue](#)

#### Declaration

**function** GetDisplayTransparentColorValue (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayTransparentColorValue(**int** DisplayIndex);

function GetDisplayTransparentColorValue (DisplayIndex as Long) as Long

#### Description

see [SetDisplayTransparentColorValue](#)

## GetDisplayVideoHeight

### TVideoGrabber.GetDisplayVideoHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

current video height

#### Declaration

**function** GetDisplayVideoHeight (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayVideoHeight(**int** DisplayIndex);

function GetDisplayVideoHeight (DisplayIndex as Long) as Long

#### Description

Returns the current video height of the video window

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## GetDisplayVideoPortEnabled

### TVideoGrabber.GetDisplayVideoPortEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayVideoPortEnabled](#)

#### Declaration

**function** GetDisplayVideoPortEnabled (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayVideoPortEnabled(**int** DisplayIndex);

function GetDisplayVideoPortEnabled ayActive (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayVideoPortEnabled](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

## GetDisplayVideoWidth

### TVideoGrabber.GetDisplayVideoWidth

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

current video width

#### Declaration

**function** GetDisplayVideoWidth (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayVideoWidth(**int** DisplayIndex);

function GetDisplayVideoWidth (DisplayIndex as Long) as Long

#### Description

Returns the current video width of the video window

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

## GetDisplayVideoWindowHandle

### TVideoGrabber.GetDisplayVideoWindowHandle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

handle of the current video window

#### Declaration

**function** GetDisplayVideoWindowHandle (DisplayIndex: LongInt): HWND;

**void \*** \_\_fastcall GetDisplayVideoWindowHandle(**int** DisplayIndex);

function GetDisplayVideoWindowHandle (DisplayIndex as Long) as Long

#### Description

retuns the handle of the current video window

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

### GetDisplayVisible

#### TVideoGrabber.GetDisplayVisible

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayVisible](#)

#### Declaration

**function** GetDisplayVisible (DisplayIndex: LongInt): Boolean;

**bool** \_\_fastcall GetDisplayVisible(**int** DisplayIndex);

function GetDisplayVisible (DisplayIndex as Long) as **bool**

#### Description

see [SetDisplayVisible](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

### GetDisplayWidth

#### TVideoGrabber.GetDisplayWidth

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

see [SetDisplayWidth](#)

#### Declaration

**function** GetDisplayWidth (DisplayIndex: LongInt): LongInt;

**int** \_\_fastcall GetDisplayWidth(**int** DisplayIndex);

function GetDisplayWidth (DisplayIndex as Long) as Long

#### Description

see [SetDisplayWidth](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

### GetFrameInfo

#### TVideoGrabber.GetFrameInfo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to retrieve information about the current video frame.

#### Declaration

**function** GetFrameInfo(FrameId: LongInt; FrameInfold: TFrameInfold): LongWord;

**unsigned** \_\_fastcall GetFrameInfo(**int** FrameId, TFrameInfold FrameInfold)



Function GetFrameInfo(FrameId As Long, FrameInfoId As TFrameInfoId) As Long

### Description

Used to retrieve information about the current video frame.

Must be invoked from the TVideoGrabber events that occurs for video frames, like [OnFrameProgress](#), [OnFrameCaptureCompleted](#), [OnMotionDetected](#), etc...

- pass as first parameter the FrameId parameter returned by the event,
- pass as second parameter the [TFrameInfoId](#) value corresponding to the desired information.

E.g.:

```
procedure TfrmMainForm.VideoGrabberFrameProgress(Sender: TObject; FrameNumber: Cardinal)
begin
    edtFrameCount.Text := 'frame: ' + IntToStr (FrameNumber)
                        + ' dropped:' + IntToStr (VideoGrabber.GetFrameInfo (FrameId, fi_DroppedFrame))
                        + ' time:' + Format ('%.2d:%.2d:%.2d %.2d', [VideoGrabber.GetFrameInfo (FrameId, fi_Time),
                                                                    VideoGrabber.GetFrameInfo (FrameId, fi_DroppedFrame),
                                                                    VideoGrabber.GetFrameInfo (FrameId, fi_FramesPerSecond),
                                                                    VideoGrabber.GetFrameInfo (FrameId, fi_FrameSize)])
end;
```

### See Also

[TFrameGrabberRGBFormat](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#) [FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfoString](#) [InFrameProgressEvent](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [WebcamStillCaptureButton](#)

---

Created with the Standard Edition of HelpNDoc: Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc

---

## GetFrameInfoString

### TVideoGrabber.GetFrameInfoString

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns information about the current frame as string

### Declaration

**function** GetFrameInfoString (FrameInfoStringId: TFrameInfoStringId): **string**;

wchar\_t \* \_\_fastcall GetFrameInfoString(TFrameInfoStringId FrameInfoStringId);

Function GetFrameInfoString (FrameInfoStringId as TFrameInfoStringId) as String

### Description

Depending of the FrameInfoStringId parameter (type [TFrameInfoStringId](#)), this function returns the following string about the current video frame:

Examples:

#### **fis\_DVTimeCode**

time code expressed in hh:mm:ss:ff, e.g. **00:15:42:08**

#### **fis\_DVDateTime**

date/time stored on the tape, e.g. **"2009-03-02 10:15:26"**

*(only for a DV camcorder playing a DV tape, this returns the date/time stored on the tape)*

### **fis\_TimeCode**

frame time expressed in hh:mm:ss:hs, e.g. **00:00:00 53**

### **fis\_FrameTime**

time expressed in 100ns units, e.g. **5351096**

### **fis\_FrameNumber**

current frame number, e.g. **17**

### **fis\_FullInfo**

all the information above in a single string, separated by (A)...(B)...(C)...(D)...(E)...(F) delimiters to facilitate an eventual parsing, e.g.:

"(A) 17 (B) 5351096 (C) 00:00:00 53 (D) 2009-03-02 10:15:35 (E) 00:15:42:08 (F)"

### **See Also**

[TCardinalDirection](#) [TAutoFileName](#) [TFrameCaptureDest](#) [TFrameGrabberRGBFormat](#)  
[TOnFrameCaptureCompleted](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#)  
[CaptureFrameTo](#) [DrawBitmapOverFrame](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#)  
[FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#)  
[FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#) [GetLastFrameAsHBITMAP](#)  
[GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#)  
[ImageOverlay](#) [StretchToVideoSize](#) [InFrameProgressEvent](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#)  
[JPEGQuality](#) [Last\\_BurstFrameCapture](#) [FileName](#) [Last\\_CaptureFrameTo\\_FileName](#)  
[MouseWheelEventEnabled](#) [OnDiskFull](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#)  
[OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#)  
[OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#)  
[SetFrameCaptureBounds](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#)  
[SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#)  
[SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#)  
[SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#)  
[SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#)  
[SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#)  
[SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#)  
[SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [StoragePath](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Create Professional Documentation with HelpNDoc's Clean UI

## **GetFWCam1394**

### **TVideoGrabber.GetFWCam1394**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves a setting and min/max/default values of Firewire/GIE camera

### **Declaration**

**function** GetFWCam1394 (FWCam1394ID: **string**; **var** Value: LongInt; **var** Flags: LongInt; **var** Capabilities: DWORD; **var** MinValue: LongInt; **var** MaxValue: LongInt; **var** Default: LongInt): Boolean;

**bool** \_\_fastcall GetFWCam1394(System::wchar\_t \*FWCam1394ID, **int** &Value, **int** &Flags, **unsigned** &Capabilities, **int** &MinValue, **int** &MaxValue, **int** &Default);

#### Description

See [SetFWCam1394](#) for sample code.

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

### GetFWCam1394List

#### TVideoGrabber.GetFWCam1394List

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the list of possible custom settings for Firewire/GIE camera

#### Declaration

**function** GetFWCam1394List: **string**;

System::wchar\_t \* \_\_fastcall GetFWCam1394List(**void**);

#### Description

Returns the list of the possible settings as a string made of text lines separated by characters

Created with the Standard Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

### GetImageOverlay\_AlphaBlend

#### TVideoGrabber.GetImageOverlay\_AlphaBlend

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_AlphaBlend](#)

#### Declaration

**function** GetImageOverlay\_AlphaBlend (**Index**: LongInt): Boolean;

**bool** \_\_fastcall GetImageOverlay\_AlphaBlend(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_AlphaBlend](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

### GetImageOverlay\_AlphaBlendValue

#### TVideoGrabber.GetImageOverlay\_AlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_AlphaBlendValue](#)

#### Declaration

**function** GetImageOverlay\_AlphaBlendValue (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_AlphaBlendValue(**int** Index);

## Description

retrieves the value set by [SetImageOverlay\\_AlphaBlendValue](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## GetImageOverlay\_ChromaKey

### TVideoGrabber.GetImageOverlay\_ChromaKey

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_ChromaKey](#)

## Declaration

**function** GetImageOverlay\_ChromaKey (**Index**: LongInt): Boolean;

**bool** \_\_fastcall GetImageOverlay\_ChromaKey(int Index);

## Description

retrieves the value set by [SetImageOverlay\\_ChromaKey](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## GetImageOverlay\_ChromaKeyLeewayPercent

### TVideoGrabber.GetImageOverlay\_ChromaKeyLeewayPercent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_ChromaKeyLeewayPercent](#)

## Declaration

**function** GetImageOverlay\_ChromaKeyLeewayPercent (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_ChromaKeyLeewayPercent(int Index);

## Description

retrieves the value set by [SetImageOverlay\\_ChromaKeyLeewayPercent](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## GetImageOverlay\_ChromaKeyRGBColor

### TVideoGrabber.GetImageOverlay\_ChromaKeyRGBColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_ChromaKeyRGBColor](#)

## Declaration

**function** GetImageOverlay\_ChromaKeyRGBColor (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_ChromaKeyRGBColor(int Index);

## Description

retrieves the value set by [SetImageOverlay\\_ChromaKeyRGBColor](#)

## GetImageOverlay\_Enabled

### TVideoGrabber.GetImageOverlay\_Enabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_Enabled](#)

#### Declaration

**function** GetImageOverlay\_Enabled (**Index**: LongInt): Boolean;

**bool** \_\_fastcall GetImageOverlay\_Enabled(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_Enabled](#)

## GetImageOverlay\_Height

### TVideoGrabber.GetImageOverlay\_Height

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_Height](#)

#### Declaration

**function** GetImageOverlay\_Height (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_Height(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_Height](#)

## GetImageOverlay\_LeftLocation

### TVideoGrabber.GetImageOverlay\_LeftLocation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_LeftLocation](#)

#### Declaration

**function** GetImageOverlay\_LeftLocation (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_LeftLocation(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_LeftLocation](#)

## GetImageOverlay\_RotationAngle

### TVideoGrabber.GetImageOverlay\_RotationAngle

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_RotationAngle](#)

#### Declaration

**function** GetImageOverlay\_RotationAngle (**Index**: LongInt): Double;

**double** \_\_fastcall GetImageOverlay\_RotationAngle(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_RotationAngle](#)

Created with the Standard Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

## GetImageOverlay\_StretchToVideoSize

### TVideoGrabber.GetImageOverlay\_StretchToVideoSize

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_StretchToVideoSize](#)

#### Declaration

**function** GetImageOverlay\_StretchToVideoSize (**Index**: LongInt): Boolean;

**bool** \_\_fastcall GetImageOverlay\_StretchToVideoSize(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_StretchToVideoSize](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

## GetImageOverlay\_TargetDisplay

### TVideoGrabber.GetImageOverlay\_TargetDisplay

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_TargetDisplay](#)

#### Declaration

**function** GetImageOverlay\_TargetDisplay(**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_TargetDisplay(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_TargetDisplay](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## GetImageOverlay\_TopLocation

### TVideoGrabber.GetImageOverlay\_TopLocation

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

## [ber](#)

retrieves the value set by [SetImageOverlay\\_TopLocation](#)

### Declaration

**function** GetImageOverlay\_TopLocation (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_TopLocation(**int** Index);

### Description

retrieves the value set by [SetImageOverlay\\_TopLocation](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## GetImageOverlay\_Transparent

### TVideoGrabber.GetImageOverlay\_Transparent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_Transparent](#)

### Declaration

**function** GetImageOverlay\_Transparent (Index: LongInt): Boolean;

**bool** \_\_fastcall GetImageOverlay\_Transparent(**int** Index);

### Description

retrieves the value set by [SetImageOverlay\\_Transparent](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## GetImageOverlay\_TransparentColorValue

### TVideoGrabber.GetImageOverlay\_TransparentColorValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_TransparentColorValue](#)

### Declaration

**function** GetImageOverlay\_TransparentColorValue (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_TransparentColorValue(**int** Index);

### Description

retrieves the value set by [SetImageOverlay\\_TransparentColorValue](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## GetImageOverlay\_UseTransparentColor

### TVideoGrabber.GetImageOverlay\_UseTransparentColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_UseTransparentColor](#)



#### Declaration

**function** GetImageOverlay\_UseTransparentColor (**Index**: LongInt): Boolean;

**bool** \_\_fastcall GetImageOverlay\_UseTransparentColor(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_UseTransparentColor](#)

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

### GetImageOverlay\_Width

#### TVideoGrabber.GetImageOverlay\_Width

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetImageOverlay\\_Width](#)

#### Declaration

**function** GetImageOverlay\_Width (**Index**: LongInt): LongInt;

**int** \_\_fastcall GetImageOverlay\_Width(**int** Index);

#### Description

retrieves the value set by [SetImageOverlay\\_Width](#)

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

### GetItemNameFromList

#### TVideoGrabber.GetItemNameFromList

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Name of a list item from index

#### Declaration

**function** GetItemNameFromList(List: **string**; ItemIndex: LongInt): **string**;

System::wchar\_t \* \_\_fastcall GetItemNameFromList(System::wchar\_t \*List, **int** ItemIndex)

Function GetItemNameFromList (List as String, ItemIndex as Long) as String

#### Description

Used to retrieve the name of an item in a text list by its index

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

### GetLastAverageStreamValue

#### TVideoGrabber.GetLastAverageStreamValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the current average stream value

#### Declaration

**function** GetLastAverageStreamValue (StreamType: TStreamType): LongInt;

**int \_\_fastcall** GetLastAverageStreamValue(TStreamType StreamType);

Function GetLastAverageStreamValue(StreamType as TStreamType)

### Description

Used to get the current average stream value, depending on [StreamType](#) :

**st\_Video**: current average value of the RGB pixels

**st\_Audio**: current average audio level

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## GetLastErrorMessage

### TVideoGrabber.GetLastErrorMessage

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

returns the last error message during the URL connection, if any

### Declaration

**function** GetLastErrorMessage: **string**;

System::UnicodeString \_\_fastcall GetLastErrorMessage();

### Description

returns the last error message that eventually occurred during the URL connection of the Datastead RTSP/RTMP/HTTP/ONVIF Source filter

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

## GetLastFrameAsHBITMAP

### TVideoGrabber.GetLastFrameAsHBITMAP

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the last video frame as a HBITMAP memory bitmap handle

### Declaration

**function** GetLastFrameAsHBITMAP (BufferIndex: LongInt; WithOverlays: Boolean; SrcLeftLocation: LongInt; SrcTopLocation: LongInt; SrcWidth: LongInt; SrcHeight: LongInt; DestWidth: LongInt; DestHeight: LongInt; BitmapColorBitCount: LongInt): HBITMAP;

HBITMAP \_\_fastcall GetLastFrameAsHBITMAP(int BufferIndex, bool WithOverlays, int SrcLeftLocation, int SrcTopLocation, int SrcWidth, int SrcHeight, int DestWidth, int DestHeight, int BitmapColorBitCount);

function GetLastFrameAsHBITMAP (BufferIndex as Long, WithOverlays as Bool, SrcLeftLocation as Long, SrcTopLocation as Long, SrcWidth as Long, SrcHeight as Long, DestWidth as Long, DestHeight as Long, BitmapColorBitCount as Long) as Long

### Description

Used to get the last video frame as a memory bitmap.

The function returns a **HBITMAP** bitmap handle.

*Don't forget to invoke **DeleteObject (bitmap handle)** to release the memory when you have done with the bitmap, otherwise you will quick run out of memory.*

**BufferIndex:** LongInt;

0 = the current video frame (usual value)

1 = the previous video frame (n-1)

2 = the video frame n-2

...

(the size of the buffer may vary, usually don't try to use frames older than n-4)

If the index is out of the buffer the function return 0

**WithOverlays:** Boolean;

if false, the native frame without overlays is returned

if true, the frame with overlays applied is returned

**SrcLeftLocation:** LongInt;

x location of the rectangle to capture (usually 0 for a full frame)

**SrcTopLocation:** LongInt;

y location of the rectangle to capture (usually 0 for a full frame)

**SrcWidth:** LongInt;

width of the rectangle to capture (usually 0 for a full width)

**SrcHeight:** LongInt;

height of the rectangle to capture (usually 0 for a full height)

**DestWidth:** LongInt;

width of the desired bitmap (0 will return a bitmap having the source width)

**DestHeight:** LongInt;

height of the desired bitmap (0 will return a bitmap having the source height)

**BitmapColorBitCount:** LongInt

color bit count of the desired bitmap. Valid values are 32, 24, 16, 15, 8.

0 returns a bitmap having the same color count than the video frame (usually 24 or 32, depending on the [FrameGrabberRGBFormat](#) setting)

E.g.:

**BitmapHandle = VideoGrabber.GetLastFrameAsTBitmap (0, false, 0, 0, 0, 0, 0, 0, 0)**

returns the last video frame without overlays in its current size

**BitmapHandle = VideoGrabber.GetLastFrameAsTBitmap (0, false, 0, 0, 0, 0, 320, 240, 8)**

returns the previous video frame with overlays as a 320x240 bitmap, 8 bits color

### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

## GetLastFrameAsTBitmap

### TVideoGrabber.GetLastFrameAsTBitmap

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Returns the last video frame as a TBitmap

#### Declaration

function GetLastFrameAsTBitmap (BufferIndex: LongInt; WithOverlays: Boolean; SrcLeftLocation: LongInt; SrcTopLocation: LongInt; SrcWidth: LongInt; SrcHeight: LongInt; DestWidth: LongInt; DestHeight: LongInt; BitmapColorBitCount: LongInt): TBitmap;

Graphics::TBitmap\* \_\_fastcall GetLastFrameAsTBitmap(int BufferIndex, bool WithOverlays, int SrcLeftLocation, int SrcTopLocation, int SrcWidth, int SrcHeight, int DestWidth, int DestHeight, int BitmapColorBitCount);

n/a

#### Description

Used to get a copy of the last video frame as a Delphi or C++Builder TBitmap (not available in the DLL and OCX versions)

See [GetLastFrameAsHBitmap](#) for the description of the parameters (it has exactly the same parameters).

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## GetLastFrameBitmapBits

### TVideoGrabber.GetLastFrameBitmapBits

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Returns a pointer to the bitmap bits of the last video frame

#### Declaration

**function** GetLastFrameBitmapBits (BufferIndex: LongInt; WithOverlays: Boolean; ReleaseFrame: Boolean): Pointer;

**void** \* \_\_fastcall GetLastFrameBitmapBits(int BufferIndex, **bool** WithOverlays, **bool** ReleaseFrame);

#### Description

Returns a pointer to the bitmap bits of the last video frame.

Invoke this function to lock and get access to the bitmap bits.

Immediately after processing the bitmap bits, invoke it again with the "ReleaseFrame" set to false to unlock the sample.

E.g.:

```
...
BYTE *pBits = GetLastFrameBitmapBits (0, true, false)
// do what you need with the bitmap bits
GetLastFrameBitmapBits (0, true, true)
...
```

Note: To get the bitmap dimensions, invoke a first time [GetLastFrameAsBitmapBits2](#).

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

---

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

---

## GetLastFrameBitmapBits2

### TVideoGrabber.GetLastFrameBitmapBits2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns a pointer to the bitmap bits of the last video frame, with the bitmap format information

#### Declaration

**function** GetLastFrameBitmapBits2 (BufferIndex: LongInt; WithOverlays: Boolean; ReleaseFrame: Boolean; **out** BitmapWidth: LongInt; **out** BitmapHeight: LongInt; **out** BitmapLineSize: LongInt; **out** BitmapSize: LongInt; **out** BitmapBitsPerPixel: LongInt): Pointer;

**void** \* \_\_fastcall GetLastFrameBitmapBits2(int BufferIndex, **bool** WithOverlays, **bool** ReleaseFrame, /\* outint &BitmapWidth, /\* outint &BitmapHeight, /\* outint &BitmapLineSize, /\* outint &BitmapSize, /\* outint &BitmapBitsPerPixel);

#### Description

Returns a pointer to the bitmap bits of the last video frame.

Invoke this function to lock and get access to the bitmap bits.

Immediately after processing the bitmap bits, invoke it again with the "ReleaseFrame" set to false to unlock the sample.

E.g.:

```
...
int BitmapWidth, BitmapHeight, BitmapLineSize, BitmapSize, BitmapBitsPerPixel;
BYTE *pBits = GetLastFrameBitmapBits2 (0, true, false, &BitmapWidth, &BitmapHeight, &BitmapLineSize, &BitmapSize, &BitmapBitsPerPixel);
CopyMemory (pDest, pBits, BitMapSize);
GetLastFrameBitmapBits (0, true, true, null, null, null, null, null);
...
```

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

---

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

---

## GetLogString

### TVideoGrabber.GetLogString

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Returns a TLogType value (returned by the OnLog event) as string

#### Declaration

**function** GetLogString (Value: TLogType): **string**;

wchar\_t \*GetLogString (TLogType Value);

Function GetLogString (Value as TLogType) as string

#### Description

Used to retrieve the string value of the [TLogType](#) parameter returned by the [OnLog](#) event.

E.g.

```
procedure TForm1.VideoGrabberLog (Sender: TObject; LogType: TLogType; Severity, InfoMsg: string);
var
    sLog: string;
begin
    sLog := VideoGrabber.GetLogString (LogType);
    Memo1.Lines.Add (sLog);
end;
```

---

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

---

## GetMiscDeviceControl

### TVideoGrabber.GetMiscDeviceControl

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Used to retrieve specific values or states on some video capture devices.

#### Declaration

**function** GetMiscDeviceControl (DeviceDataType: TMiscDeviceControl; **Index**: LongInt): LongInt;

**int \_\_fastcall** GetMiscDeviceControl(TMiscDeviceControl MiscDeviceControl, **int** Index);

Function GetMiscDeviceControl (MiscDeviceControl as TMiscDeviceControl, Index as Long) as Long

#### Description

Used to retrieve specific values or states on some video capture devices that support them, e.g. the GPIO, VPD, etc...

The use of this function is explained in the [Miscellaneous device control](#) chapter.

#### See Also

---

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

---

## GetNearestVideoHeight

### TVideoGrabber.GetNearestVideoHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the nearest video height to a given video width and height.

#### Declaration

**function** GetNearestVideoHeight(PreferredVideoWidth: LongInt; PreferredVideoHeight: LongInt): LongInt;

**int** \_\_fastcall GetNearestVideoHeight(**int** PreferredVideoWidth, **int** PreferredVideoHeight)

Function GetNearestVideoHeight(PreferredVideoWidth As Long, PreferredVideoHeight As Long) As Long

#### Description

Retrieves the nearest video height to a given video width and height, available on the [current video capture device](#) .

Unlike [UseNearestVideoSize](#) , this function simply request the nearest height available for the specified size, without selecting it on the [current video capture device](#) .

#### See Also

[GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## GetNearestVideoSize

### TVideoGrabber.GetNearestVideoSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the nearest video size to a given video width and height.

#### Declaration

**procedure** GetNearestVideoSize(PreferredVideoWidth: LongInt; PreferredVideoHeight: LongInt; **out** NearestVideoWidth: LongInt; **out** NearestVideoHeight: LongInt);

**void** \_\_fastcall GetNearestVideoSize(**int** PreferredVideoWidth, **int** PreferredVideoHeight, **int** &NearestVideoWidth, **int** &NearestVideoHeight)

Sub GetNearestVideoSize(PreferredVideoWidth As Long, PreferredVideoHeight As Long, Nearest\_VideoWidth, Nearest\_VideoHeight)

#### Description

Retrieves the nearest video size to a given video width and height, available on the [current video capture device](#) .

Unlike [UseNearestVideoSize](#) , this function simply request the nearest video size, without selecting it on the [current video capture device](#) .

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)



## GetNearestVideoWidth

### TVideoGrabber.GetNearestVideoWidth

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Retrieves the nearest video width to a given video width and height.

#### Declaration

**function** GetNearestVideoWidth(PreferredVideoWidth: LongInt; PreferredVideoHeight: LongInt): LongInt;

**int** \_\_fastcall GetNearestVideoWidth(**int** PreferredVideoWidth, **int** PreferredVideoHeight)

Function GetNearestVideoWidth(PreferredVideoWidth As Long, PreferredVideoHeight As Long) As Long

#### Description

Retrieves the nearest video width to a given video width and height, available on the [current video capture device](#) .

Unlike [UseNearestVideoSize](#) , this function simply request the nearest width available for the specified size, without selecting it on the [current video capture device](#) .

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

## GetPixelsDistance

### TVideoGrabber.GetPixelsDistance

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Returns the distance between 2 pixels

#### Declaration

**function** GetPixelsDistance (x1: LongInt; y1: LongInt; x2: LongInt; y2: LongInt): Double;

**double** \_\_fastcall GetPixelsDistance(**int** x1, **int** y1, **int** x2, **int** y2);

Function GetPixelsDistance(x1 As Long, y1 As Long, x2 as Long, y2 as Long) As Double

#### Description

Used to retrieve the distance, in pixels, between a pixel located at the (x1,y1) coordinate and a pixel located at the (x2,y2) coordinate

#### See Also

[Image overlays](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)

[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## GetPlaylist

### TVideoGrabber.GetPlaylist

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the current playlist.

#### Declaration

**function** GetPlaylist: string;

wchar\_t \*GetPlaylist();

Function GetPlaylist as string

#### Description

Returns the current playlist as a list of strings separated by CR/LF characters ( char (13) and char(10) )

See the "[Using the playlist](#)" chapter for more information about the Playlist feature.

#### See Also

[TPlaylist](#) [Video formats](#) [IsPlaylistActive](#) [OnPlayerEndOfPlaylist](#) [Playlist](#) [PlaylistIndex](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

## GetRGBPixelAt

### TVideoGrabber.GetRGBPixelAt

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the RGB value of the specified pixel.

#### Declaration

**function** GetRGBPixelAt (x: LongInt; y: LongInt): Longint;

**int** GetRGBPixelAt(**int** x, **int** y);

Function GetRGBPixelAt (x as Long, y as Long) as Long

#### Description

Used to retrieve the RGB value of the specified pixel at the (x, y) location in the current video frame.

Note: this function must be invoked from one of the OnFrameOverlayUsing... events (e.g. from the [OnFrameOverlayUsingDC](#) event).

This function returns the RGB value at the x, y location on the video frame, with:  
x between 0 and (VideoSourceWidth -1)  
and  
y between 0 and <= (VideoSourceHeight - 1)

You can extract each pixel color from the value returned by using the following sample code:

#### DELPHI

```

procedure TfrmMainForm.VideoGrabberFrameOverlayUsingDC(Sender: TObject;
  Dc: HDC; FrameNumber: Cardinal; FrameTime: Int64; FrameId: Integer);
var
  RGBValue: LongInt;
  R, G, B: LongInt;
  x, y: LongInt;
begin
  x := 10; // e.g. row 10
  y := 15; // and column 15
  RGBValue := VideoGrabber.GetRGBPixelAt (x, y);
  R := RGBValue and $FF;
  G := (RGBValue shr 8) and $FF;
  B := (RGBValue shr 16) and $FF;
end;

```

## C++

```

void __fastcall TfrmMainForm::VideoGrabberFrameOverlayUsingDC(
  TObject *Sender, HDC Dc, DWORD FrameNumber, Int64 FrameTime,
  int FrameId)

  int RGBValue, R, G, B, x, y;

  x = 10; // e.g. row 10
  y = 15; // and column 15

  RGBValue = VideoGrabber->GetRGBPixelAt (x, y);

  R = RGBValue && 0xFF;
  G = (RGBValue >> 8) && 0xFF;
  B = (RGBValue >> 16) && 0xFF;

```

## C#

```

private void axVideoGrabberNET1_OnFrameOverlayUsingDC(object sender, Axvidgrab_NET.IVideoGrabber
  ...
  Color RGBColor = System.Drawing.Color.FromArgb (axVideoGrabberNET1.GetRGBPixelAt(10, 15));
  textBoxR.Text = RGBColor.R.ToString();
  textBoxG.Text = RGBColor.G.ToString();
  textBoxB.Text = RGBColor.B.ToString();
  ...

```

## See Also

[Image overlays](#) [GetPixelsDistance](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

## GetTextOverlay\_Align

### TVideoGrabber.GetTextOverlay\_Align

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Align](#)

#### Declaration

function GetTextOverlay\_Align(Index: LongInt): TTextOverlayAlign;

TTextOverlayAlign \_\_fastcall GetTextOverlay\_Align(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Align](#)

Created with the Standard Edition of HelpNDoc: [Easily share your documentation with the world through a beautiful website](#)

## GetTextOverlay\_AlphaBlend

### TVideoGrabber.GetTextOverlay\_AlphaBlend

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_AlphaBlend](#)

#### Declaration

function GetTextOverlay\_AlphaBlend (Index: LongInt): Boolean;

bool \_\_fastcall GetTextOverlay\_AlphaBlend(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_AlphaBlend](#)

#### See Also

[GetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#)  
[SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#)  
[SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_VideoAlignment](#)

Created with the Standard Edition of HelpNDoc: [HelpNDoc's Project Analyzer: Incredible documentation assistant](#)

## GetTextOverlay\_AlphaBlendValue

### TVideoGrabber.GetTextOverlay\_AlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_AlphaBlendValue](#)

#### Declaration

function GetTextOverlay\_AlphaBlendValue (Index: LongInt): LongInt;

**int** \_\_fastcall GetTextOverlay\_AlphaBlendValue(**int** Index);

#### Description

retrieves the value set by [SetTextOverlay\\_AlphaBlendValue](#)

#### See Also

[GetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#)  
[SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#)  
[SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_VideoAlignment](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

### GetTextOverlay\_BkColor

#### TVideoGrabber.GetTextOverlay\_BkColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_BkColor](#)

#### Declaration

function GetTextOverlay\_BkColor(Index: LongInt): TColor;

Graphics::TColor \_\_fastcall GetTextOverlay\_BkColor(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_BkColor](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

### GetTextOverlay\_Enabled

#### TVideoGrabber.GetTextOverlay\_Enabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Enabled](#)

#### Declaration

function GetTextOverlay\_Enabled(Index: LongInt): Boolean;

bool \_\_fastcall GetTextOverlay\_Enabled(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Enabled](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

### GetTextOverlay\_Font

#### TVideoGrabber.GetTextOverlay\_Font

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Font](#)

#### Declaration

function GetTextOverlay\_Font(Index: LongInt): TFont;

Graphics::TFont\* \_\_fastcall GetTextOverlay\_Font(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Font](#)

Created with the Standard Edition of HelpNDoc: [Experience a User-Friendly Interface with HelpNDoc's Documentation Tool](#)

### GetTextOverlay\_GradientColor

#### TVideoGrabber.GetTextOverlay\_GradientColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_GradientColor](#)

#### Declaration

function GetTextOverlay\_GradientColor(Index: LongInt): TColor;

Graphics::TColor \_\_fastcall GetTextOverlay\_GradientColor(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_GradientColor](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

### GetTextOverlay\_GradientMode

#### TVideoGrabber.GetTextOverlay\_GradientMode

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_GradientMode](#)

#### Declaration

function GetTextOverlay\_GradientMode(Index: LongInt): TTextOverlayGradientMode;

TTextOverlayGradientMode \_\_fastcall GetTextOverlay\_GradientMode(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_GradientMode](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

### GetTextOverlay\_HighResFont

#### TVideoGrabber.GetTextOverlay\_HighResFont

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_HighResFont](#)

#### Declaration

function GetTextOverlay\_HighResFont(Index: LongInt): Boolean;

bool \_\_fastcall GetTextOverlay\_HighResFont(int Index);

## Description

retrieves the value set by [SetTextOverlay\\_HighResFont](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Word Document into a Professional eBook with HelpNDoc](#)

## GetTextOverlay\_Left

### TVideoGrabber.GetTextOverlay\_Left

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Left](#)

#### Declaration

function GetTextOverlay\_Left(Index: LongInt): LongInt;

int \_\_fastcall GetTextOverlay\_Left(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Left](#)

Created with the Standard Edition of HelpNDoc: [Free PDF documentation generator](#)

## GetTextOverlay\_Right

### TVideoGrabber.GetTextOverlay\_Right

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Right](#)par

#### Declaration

function GetTextOverlay\_Right(Index: LongInt): LongInt;

int \_\_fastcall GetTextOverlay\_Right(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Right](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## GetTextOverlay\_Scrolling

### TVideoGrabber.GetTextOverlay\_Scrolling

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Scrolling](#)par

#### Declaration

function GetTextOverlay\_Scrolling(Index: LongInt): Boolean;

bool \_\_fastcall GetTextOverlay\_Scrolling(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Scrolling](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)



## GetTextOverlay\_ScrollingSpeed

### TVideoGrabber.GetTextOverlay\_ScrollingSpeed

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_ScrollingSpeed](#)

#### Declaration

function GetTextOverlay\_ScrollingSpeed(Index: LongInt): LongInt;

int \_\_fastcall GetTextOverlay\_ScrollingSpeed(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_ScrollingSpeed](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Documents to eBooks: A Step-by-Step Guide with HelpNDoc](#)

## GetTextOverlay\_Shadow

### TVideoGrabber.GetTextOverlay\_Shadow

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Shadow](#)

#### Declaration

function GetTextOverlay\_Shadow(Index: LongInt): Boolean;

bool \_\_fastcall GetTextOverlay\_Shadow(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_Shadow](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly bring your documentation online with HelpNDoc](#)

## GetTextOverlay\_ShadowColor

### TVideoGrabber.GetTextOverlay\_ShadowColor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_ShadowColor](#)

#### Declaration

function GetTextOverlay\_ShadowColor(Index: LongInt): TColor;

Graphics::TColor \_\_fastcall GetTextOverlay\_ShadowColor(int Index);

#### Description

retrieves the value set by [SetTextOverlay\\_ShadowColor](#)

Created with the Standard Edition of HelpNDoc: [Easily create CHM Help documents](#)

## GetTextOverlay\_ShadowDirection

### TVideoGrabber.GetTextOverlay\_ShadowDirection

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

## [ber](#)

retrieves the value set by [SetTextOverlay\\_ShadowDirection](#)

### Declaration

function GetTextOverlay\_ShadowDirection(Index: LongInt): TCardinalDirection;

TCardinalDirection \_\_fastcall GetTextOverlay\_ShadowDirection(int Index);

### Description

retrieves the value set by [SetTextOverlay\\_ShadowDirection](#)

Created with the Standard Edition of HelpNDoc: [Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

## GetTextOverlay\_String

### TVideoGrabber.GetTextOverlay\_String

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_String](#)

### Declaration

function GetTextOverlay\_String(Index: LongInt): string;

wchar\_t \*\_\_fastcall GetTextOverlay\_String(int Index);

### Description

retrieves the value set by [SetTextOverlay\\_String](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

## GetTextOverlay\_TargetDisplay

### TVideoGrabber.GetTextOverlay\_TargetDisplay

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_TargetDisplay](#)

### Declaration

function GetTextOverlay\_TargetDisplay(Index: LongInt): LongInt;

int \_\_fastcall GetTextOverlay\_TargetDisplay(int Index);

### Description

retrieves the value set by [SetTextOverlay\\_TargetDisplay](#)

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## GetTextOverlay\_Top

### TVideoGrabber.GetTextOverlay\_Top

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Top](#)

### Declaration

```
function GetTextOverlay_Top(Index: LongInt): LongInt;
```

```
int __fastcall GetTextOverlay_Top(int Index);
```

#### Description

retrieves the value set by [SetTextOverlay\\_Top](#)

---

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

---

## GetTextOverlay\_Transparent

### TVideoGrabber.GetTextOverlay\_Transparent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

retrieves the value set by [SetTextOverlay\\_Transparent](#)

#### Declaration

```
function GetTextOverlay_Transparent(Index: LongInt): Boolean;
```

```
bool __fastcall GetTextOverlay_Transparent(int Index);
```

#### Description

retrieves the value set by [SetTextOverlay\\_Transparent](#)

---

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

---

## GetTranslatedCoordinates

### TVideoGrabber.GetTranslatedCoordinates

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to retrieve the translated coordinates of the mouse

#### Declaration

```
function GetTranslatedCoordinates (DisplayIndex: LongInt; NativeX: LongInt; NativeY: LongInt; var TranslatedX: LongInt; var TranslatedY: LongInt): Boolean;
```

```
__fastcall GetTranslatedCoordinates(int DisplayIndex, int NativeX, int NativeY, int &TranslatedX, int &TranslatedY);
```

Function GetTranslatedCoordinates (DisplayIndex as Long, NativeX as Long, NativeY as Long, ByRef TranslatedX as Long, ByRef TranslatedY as Long) as Boolean

#### Description

Invoke GetTranslatedCoordinates to retrieve the translated coordinates (the coordinates into the video window, left/top corner = 0,0 ) of the mouse events when [TranslateMouseCoordinates](#) is disabled.

This function can be invoked from the following events:

[OnMouseDown](#)

[OnMouseUp](#)

[OnMouseMove](#)

[OnMouseWheel](#)

[OnDragOver](#)

[OnDragDrop](#)

---

Created with the Standard Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

---

## GetTVChannelInfo

### TVideoGrabber.GetTVChannelInfo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns information about the current TV channel selected.

#### Declaration

**function** GetTVChannelInfo(Value: TTVChannelInfo): LongInt;

**int** \_\_fastcall GetTVChannelInfo(TTVChannelInfo Value)

Function GetTVChannelInfo(param1 As TxTVChannelInfo) As Long

#### Description

Used to get information about the current TV channel selected.

Conditions:

- the [current video capture device](#) must have a TV tuner ([IsTVTunerAvailable](#) must return "true")
- the [VideoInput](#) property must be set to the index of "tuner" in the [VideoInputs](#) list.
- the preview or recording must be running.

Pass to this function the [TTVChannelInfo](#) value TTVChannelInfo desired, e.g.:

```
CurrentDefaultFreq = VideoGrabber.GetTVChannelInfo (tci_DefaultVideoFrequency)
```

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## GetVideoCodec

### TVideoGrabber.GetVideoCodec

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Current video codec

#### Declaration

**function** GetVideoCodec: **string**;

**System::wchar\_t \***\_\_fastcall GetVideoCodec()

Function GetVideoCodec as String

#### Description

Returns the current video codec beeing used.

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

## GetVideoCompressionSettings

### TVideoGrabber.GetVideoCompressionSettings

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the video compression settings of certain codecs.

#### Declaration

**function** GetVideoCompressionSettings(**out** DataRate, KeyFrameRate, PFramesPerKeyFrame, WindowSize: LongInt; **out** Quality: Double; **out** CanQuality, CanCrunch, CanKeyFrame, CanBFrame, CanWindow: Boolean): Boolean;

**bool** \_\_fastcall GetVideoCompressionSettings(**int** &DataRate, **int** &KeyFrameRate, **int** &PFramesPerKeyFrame, **int** &WindowSize, **double** &Quality, **bool** &CanQuality, **bool** &CanCrunch, **bool** &CanKeyFrame, **bool** &CanBFrame, **bool** &CanWindow)

Function GetVideoCompressionSettings(DataRate, KeyFrameRate, PFramesPerKeyFrame, WindowSize, Quality, CanQuality, CanCrunch, CanKeyFrame, CanBFrame, CanWindow) As Boolean

#### Description

Retrieves the video compression settings supported by some codecs. Returns true if the current video compressor supports this interface.

See [SetVideoCompressionSettings](#) .

**DataRate:** retrieves the output data rate.

**KeyFrameRate:** The key-frame rate is the number of frames per key frame. For example, if the rate is 15, then a key frame occurs every 15 frames.

**PFramesPerKeyFrame:** P frames are used only in MPEG compression. E.g. let's say a key frame occurs once every 10 frames, and there are three P frames per key frame. The P frames will be spaced evenly between the key frames. The remaining six frames are bi-directional (B) frames.

**WindowSize:** retrieves the number of frames over which the compressor will maintain the average data rate. E.g. if a data rate of 100K/sec and a frame rate of 10 frames per second, if the window size is 1, then every frame will be 10K or less. If the window size is 5, then every five consecutive frames will average 10K per frame, but individual frames may exceed this size.

**Quality:** The quality is expressed as a value between 0.0 and 1.0, where 1.0 indicates the best quality and 0.0 indicates the worst quality. If the value is negative, the filter will use the default quality.

**CanCrunch:** the compressor can compress video to a specified data rate (see DataRate above).

**CanKeyFrame:** the compressor supports the KeyFrame property above.

**CanBFrame:** the compressor supports the PFramesPerKeyFrame property above.

**CanWindow:** the compressor supports WindowSize property above.

**CanQuality:** the compressor supports Quality property above.

#### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## GetVideoControlMode

### TVideoGrabber.GetVideoControlMode

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieve the state of one of the [TVideoControl](#) settings

#### Declaration

**function** GetVideoControlMode (Mode: TVideoControl): Boolean;

**bool** \_\_fastcall GetVideoControlMode(TVideoControl Mode);

Function GetVideoControlMode (mode as TVideoControl) As Boolean

#### Description

Used to retrieve the state of one of the [TVideoControl](#) settings, if available for the current video capture device (see [IsVideoControlModeAvailable](#) ).

#### See Also

[TVideoControl](#) [IsVideoControlAvailable](#) [IsVideoControlModeAvailable](#) [SetVideoControlMode](#)  
[SetVideoControlMode2](#) [VideoControlSettings](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Documentation Process with HelpNDoc's Project Analyzer](#)

## GetVideoHeightFromIndex

### TVideoGrabber.GetVideoHeightFromIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the height of a video size in the [VideoSizes](#) list.

#### Declaration

**function** GetVideoHeightFromIndex(VideoSizeIndex: LongInt): LongInt;

**int** \_\_fastcall GetVideoHeightFromIndex(int VideoSizeIndex)

Function GetVideoHeightFromIndex(VideoSizeIndex As Long) As Long

#### Description

Used to retrieve the Height of a video size in the [VideoSizes](#) list of the [current video capture device](#) . The VideoSizeIndex parameter is the index of the size in the [VideoSizes](#) list. Returns 0 is VideoSizeIndex is out of the 0..[VideoSizesCount](#) - 1 range.

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoSizeFromIndex](#)  
[GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#)  
[VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## GetVideoSizeFromIndex

### TVideoGrabber.GetVideoSizeFromIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the width and height of a video size in the [VideoSizes](#) list.

#### Declaration

**function** GetVideoSizeFromIndex(VideoSizeIndex: LongInt; **out** VideoWidth: LongInt; **out** VideoHeight: LongInt): Boolean;

**bool** \_\_fastcall GetVideoSizeFromIndex(**int** VideoSizeIndex, **int** &VideoWidth, **int** &VideoHeight)

Function GetVideoSizeFromIndex(VideoSizeIndex As Long, Video\_Width, Video\_Height) As Boolean

#### Description

Used to retrieve the width and height of a video size in the [VideoSizes](#) list of the [current video capture device](#) .

The VideoSizeIndex parameter is the index of the size in the [VideoSizes](#) list.

The width and height are returned by the VideoWidth and VideoHeight parameters.

Returns false is VideoSizeIndex is out of the 0..[VideoSizesCount](#) - 1 range.

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#)  
[GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#)  
[VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## GetVideoWidthFromIndex

### TVideoGrabber.GetVideoWidthFromIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the width of a video size in the [VideoSizes](#) list.

#### Declaration

**function** GetVideoWidthFromIndex(VideoSizeIndex: LongInt): LongInt;

**int** \_\_fastcall GetVideoWidthFromIndex(**int** VideoSizeIndex)

Function GetVideoWidthFromIndex(VideoSizeIndex As Long) As Long

#### Description

Used to retrieve the width of a video size in the [VideoSizes](#) list of the [current video capture device](#) .

The VideoSizeIndex parameter is the index of the size in the [VideoSizes](#) list.

Returns 0 is VideoSizeIndex is out of the 0..[VideoSizesCount](#) - 1 range.

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#)  
[GetVideoSizeFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#)  
[VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## GetVMR9ImageAdjustmentBounds

### TVideoGrabber.GetVMR9ImageAdjustmentBounds

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the bounds of the specified VMR9 image adjustment property.

#### Declaration

**function** GetVMR9ImageAdjustmentBounds(MainDisplay: Boolean; VMR9ControlSetting: TVMR9ImageAdjustment; **out** MinValue: LongInt; **out** MaxValue: LongInt; **out** StepSize: LongInt; **out** DefaultValue: LongInt; **out** CurrentValue: LongInt): Boolean;

**bool** \_\_fastcall GetVMR9ImageAdjustmentBounds(**bool** MainDisplay, TVMR9ImageAdjustment



VMR9ControlSetting, **int** &MinValue, **int** &MaxValue, **int** &StepSize, **int** &DefaultValue, **int** &CurrentValue)

Function GetVMR9ImageAdjustmentBounds(MainDisplay As Boolean, VMR9ControlSetting As TxVMR9ImageAdjustment, MinValue, MaxValue, StepSize, DefaultValue, CurrentValue) As Boolean

### Description

Used to retrieve the bounds of the specified VMR9 image adjustment property (brightness, contrast, hue, saturation).

The VMR9ControlSetting parameter is a [TVMR9ImageAdjustment](#) type.

Returns all the information useful to setup a trackbar component (min value, max value, step size, default value, current position).

E.g.:

```
procedure TForm1.VideoGrabberGraphBuilt(Sender: TObject);
var
    MinValue, MaxValue, StepSize, DefaultValue, CurrentValue: LongInt;
begin
    BrightnessTrackbar.Enabled := VideoGrabber.GetVMR9ImageAdjustmentBounds (True, vmr9_Brightness, MinValue, MaxValue, StepSize, DefaultValue, CurrentValue);
    if Brightness.Enabled then begin
        Brightness.Min := MinValue;
        Brightness.Max := MaxValue;
        Brightness.Frequency := StepSize;
        Brightness.Position := CurrentValue;
    end;
end;

procedure TForm1.BrightnessTrackbarChange(Sender: TObject);
begin
    VideoGrabber.SetVMR9ImageAdjustmentValue (True, vmr9_Brightness, tbrVMR9Brightness.Position);
end;
```

See the MainDemo project for more sample code.

### See Also

[IsVMR9ImageAdjustmentAvailable](#) [SetVMR9ImageAdjustmentValue](#) [TVMR9ImageAdjustment](#)

---

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

---

## GetVUMeterSetting

### TVideoGrabber.GetVUMeterSetting

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves a current VUMeter setting

### Declaration

**function** GetVUMeterSetting(ChannelIndex: LongWord; VUMeterSetting: TVUMeterSetting): LongWord;

**unsigned \_\_fastcall** GetVUMeterSetting(**unsigned** ChannelIndex, TVUMeterSetting VUMeterSetting)

Function GetVUMeterSetting (ChannelIndex as Long, VUMeterSetting as TVUMeterSetting) as Long

### Description

Used to retrieve a current VUMeter setting.

See [SetVUMeterSetting](#)

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

## GraphState

### TVideoGrabber.GraphState

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

returns the current state of the graph

#### Declaration

**function** GraphState: TGraphState;

TGraphState \_\_fastcall GetGraphState(void)

Function GraphState as TxGraphState

#### Description

Used to retrieve the current state of the preview, recording or playback graph.

The function returns a [TGraphState](#) value.

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

## IsAudioDeviceASoundCard

### TVideoGrabber.IsAudioDeviceASoundCard

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Properties](#)

Returns true if the audio device specified is a sound card.

#### Declaration

**function** IsAudioDeviceASoundCard (DeviceIndex: LongInt): Boolean;

bool \_\_fastcall IsAudioDeviceASoundCard(int DeviceIndex)

Function IsAudioDeviceASoundCard (DeviceIndex as Long) As Boolean

#### Description

Used to know if the [audio device](#) specified is a sound card.

Specify as DeviceIndex the index of the audio capture device in the [AudioDevices](#) list.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Easy EBook and documentation generator](#)

## IsAudioDeviceConnected

### TVideoGrabber.IsAudioDeviceConnected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to know if the current audio capture device is connected

#### Declaration

**function** IsAudioDeviceConnected(DeviceIndex: LongInt): Boolean;

**bool \_\_fastcall** IsAudioDeviceConnected(**int** DeviceIndex);

Function IsAudioDeviceConnected (DeviceIndex As Long) As Boolean

#### Description

Returns true if the specified audio capture device (in the [AudioDevices](#) list) is connected.  
Returns false if the device has been disconnected.

See the [Audio capture devices](#) chapter for more information.

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

## IsAudioRendererConnected

### TVideoGrabber.IsAudioRendererConnected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns true when an audio renderer is connected

#### Declaration

**function** IsAudioRendererConnected(RendererIndex: LongInt): Boolean;

**bool \_\_fastcall** IsAudioRendererConnected(**int** RendererIndex)

Function IsAudioRendererConnected (RendererIndex as Long) as Boolean

#### Description

Used to know if the audio renderer is connected.  
Specify as parameter the index of the audio renderer in the [AudioRenderers](#) list.

#### See Also

[AudioRendererIndex](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## IsCameraControlSettingAvailable

### TVideoGrabber.IsCameraControlSettingAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Availability of a given [TCameraControl](#) setting.

#### Declaration

**function** IsCameraControlSettingAvailable(Setting: TCameraControl): Boolean;

**bool** \_\_fastcall IsCameraControlSettingAvailable(TCameraControl Setting)

Function IsCameraControlSettingAvailable(Setting As TxCameraControl) As Boolean

#### Description

Retrieves the availability of a given [TCameraControl](#) setting for the current video capture device.

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [SendCameraCommand](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## IsDialogAvailable

### TVideoGrabber.IsDialogAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Availability of a given [TDialog](#) .

#### Declaration

**function** IsDialogAvailable(Dialog: TDialog): Boolean;

**bool** \_\_fastcall IsDialogAvailable(TDialog Dialog)

Function IsDialogAvailable(Dialog As TxDialog) As Boolean

#### Description

Retrieves the availability of a given [TDialog](#) for the current video capture device.

Created with the Standard Edition of HelpNDoc: [Effortlessly Edit and Export Markdown Documents](#)

## IsDVDevice

### TVideoGrabber.IsDVDevice

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to know if the specified video capture device is a DV device.

#### Declaration

**function** IsDVDevice(Index: LongInt): Boolean;

**bool** \_\_fastcall IsDVDevice(int Index);

Function IsDVDevice(Index As Long) As Boolean

#### Description

Used to know if the video capture device specified by its index [VideoDevices](#) list is a DV capture device.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your CHM Help Files with HelpNDoc's Advanced](#)

## Customization Options

### IsPlaylistActive

#### TVideoGrabber.IsPlaylistActive

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the state of the playlist.

#### Declaration

**function** IsPlaylistActive: Boolean;

**bool** IsPlaylistActive(**void**);

Function IsPlaylistActive as Boolean

#### Description

Returns true if the playlist is active (playing).

See the "[Using the playlist](#)" chapter for more information about the playlist feature.

#### See Also

[TPlaylist](#) [Video formats](#) [GetPlaylist](#) [OnPlayerEndOfPlaylist](#) [Playlist](#) [PlaylistIndex](#)

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

### IsURLResponding

#### TVideoGrabber.IsURLResponding

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns true if the URL is alive

#### Declaration

**function** IsURLResponding: Boolean;

**bool** \_\_fastcall IsURLResponding(**void**);

#### Description

Used to verify if the URL specified to [IPCameraURL](#) or [VideoSource\\_FileOrURL](#) without starting to decode the live stream.

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Easily Add Encryption and Password Protection to Your PDFs](#)

### IsURLVideoStreamAvailable

#### TVideoGrabber.IsURLVideoStreamAvailable

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

used to determine if the URL source really outputs video samples

### Declaration

**function** IsURLVideoStreamAvailable (TimeOut\_Ms: LongInt): TTriState;

**TTriState** \_\_fastcall IsURLVideoStreamAvailable (int TimeOut\_Ms);

### Description

Used to verify if the URL specified to [IPCameraURL](#) or [VideoSource\\_FileOrURL](#) outputs video samples without starting the GUI.

This function can be invoked from a background thread.

The function waits for the first video sample to be received, until the Timeout\_Ms duration (expressed in milliseconds) expires.

Return values:

**ts\_Undefined:** the connection failed

**ts\_False:** the connection succeeded but the source does not output video samples

**ts\_True:** the connection succeeded and at least one video sample has been received

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Edit and Export Markdown Documents](#)

## IsVideoControlModeAvailable

### [TVideoGrabber](#).IsVideoControlModeAvailable

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Methods](#)

Availability of a given [TVideoControl](#) mode.

### Declaration

**function** IsVideoControlModeAvailable(Mode: TVideoControl): Boolean;

**bool** \_\_fastcall IsVideoControlModeAvailable(TVideoControl Mode)

Function IsVideoControlModeAvailable(Mode As TxVideoControl) As Boolean

### Description

Retrieves the availability of a given [TVideoControl](#) mode for the current video capture device.

### See Also

[TVideoControl](#) [GetVideoControlMode](#) [IsVideoControlAvailable](#) [SetVideoControlMode](#) [SetVideoControlMode2](#) [VideoControlSettings](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

## IsVideoDeviceConnected

### [TVideoGrabber](#).IsVideoDeviceConnected

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Methods](#)

Returns true if the specified video capture device is connected.

#### Declaration

**function** IsVideoDeviceConnected(DeviceIndex: LongInt): Boolean;

**bool** \_\_fastcall IsVideoDeviceConnected(int DeviceIndex);

Function IsVideoDeviceConnected (DeviceIndex As Long) As Boolean

#### Description

Returns true if the video capture device specified by its index (in the [VideoDevices](#) list) is connected.  
Returns false if the device has been disconnected.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#)  
[FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#)  
[IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#)  
[ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#)  
[VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## IsVideoQualitySettingAvailable

### TVideoGrabber.IsVideoQualitySettingAvailable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Availability of a given [TVideoQuality](#) setting.

#### Declaration

**function** IsVideoQualitySettingAvailable(Setting: TVideoQuality): Boolean;

**bool** \_\_fastcall IsVideoQualitySettingAvailable(TVideoQuality Setting)

Function IsVideoQualitySettingAvailable(Setting As TxVideoQuality) As Boolean

#### Description

Retrieves the availability of a given [TVideoQuality](#) setting for the current video capture device.

#### See Also

[IsVideoQualityAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#) [VideoQualityDefault](#) [VideoQualityMax](#)  
[VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## IsVideoSignalDetected

### TVideoGrabber.IsVideoSignalDetected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Detects if a video signal is present.

#### Declaration

**function** IsVideoSignalDetected (DetectConnexantBlueScreen: **Boolean**; DetectCustomRGB: **Boolean**;  
CustomR, CustomG, CustomB: LongInt; UseAsMaxValues: Boolean): **Boolean**;



**bool** \_\_fastcall IsVideoSignalDetected (**bool** DetectConnexantBlueScreen, **bool** DetectCustomRGB, **int** CustomR, **int** CustomG, **int** CustomB, **bool** UseAsMaxValues)

Function IsVideoSignalDetected (DetectConnexantBlueScreen as Boolean; DetectCustomRGB as Boolean, CustomR as Long, CustomG as Long, CustomB as Long, UseAsMaxValues as Boolean) as Boolean

### Description

**Used to detect if a video signal is present.**

It returns:

- true if a video signal is detected,
- false if a blue screen or dark screen is detected (see below).

This function can act in 2 ways, depending of the DetectConnexantBlueScreen and DetectCustomRGB parameters (that can be combined):

**DetectConnexantBlueScreen** is enabled:

The "classical blue screen" of video capture boards including Connexant video chips, that corresponds to "no video signal" is detected (in this case the function will return false)

**DetectCustomRGB** is enabled:

The CustomR, CustomG and CustomB parameters specify the values of the colors used to detect if a video signal is present, according to the **UseAsMaxValues** parameter:

**UseAsMaxValues is disabled:**

The video signal will be reported as "not detected" if all the pixels tested on the video frames have the exact RGB values specified in CustomR, CustomG and CustomB.

**UseAsMaxValues is enabled:**

The video signal will be reported as "not detected" if all the pixels tested on the video frames have a lower value than the CustomR, CustomG and CustomB values specified.

E.g:

Connexant (BT8x8) cards:

**IsVideoSignalDetected (true, false, 0, 0, 0, false)** returns false when the blue screen of **Conexant BT8x8** based cards is detected.

Techwell cards:

**IsVideoSignalDetected (false, true, 16, 16, 16, false)** returns false when the black screen of **Techwell** cards is detected.

Conexant as well as Techwell cards

**IsVideoSignalDetected (true, true, 16, 16, 16, false)** returns false when either the Conexant's blue screen or the Techwell's black screen is detected.

True black video

**IsVideoSignalDetected (false, true, 0, 0, 0, false)** returns false on a really black video (all RGB values = 0)

Detection of a dark ambient light

**IsVideoSignalDetected (false, true, 30, 30, 30, true)** returns false on a dark or black video (all RGB values below 30)

---

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

---

## IsVMR9ImageAdjustmentAvailable

**TVideoGrabber.IsVMR9ImageAdjustmentAvailable**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the availability of a VMR9 image adjustment setting.

## Declaration

**function** IsVMR9ImageAdjustmentAvailable(MainDisplay: Boolean): Boolean;

**bool** \_\_fastcall IsVMR9ImageAdjustmentAvailable(**bool** MainDisplay)

Function IsVMR9ImageAdjustmentAvailable(MainDisplay As Boolean) As Boolean

## Description

Used to retrieve the availability of a VMR9 image adjustment property (brightness, contrast, hue, saturation).

## See Also

[GetVMR9ImageAdjustmentBounds](#) [SetVMR9ImageAdjustmentValue](#) [TVMR9ImageAdjustment](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## LoadCompressorSettingsFromDataString

**TVideoGrabber.LoadCompressorSettingsFromDataString**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Reloads the compressor settings from a data string

## Declaration

**function** LoadCompressorSettingsFromDataString (IsVideoCompressor: boolean; CompressorIndex: LongInt; DataString: string): **Boolean**;

**bool** \_\_fastcall LoadCompressorSettingsFromDataString (**bool** IsVideoCompressor, **int** CompressorIndex: LongInt, wchar\_t \*DataString);

function LoadCompressorSettingsFromDataString (IsVideoCompressor as Boolean, CompressorIndex as Long, FileName as String) As String

## Description

Used to reload the compressor settings from a data string.

The compressor settings depend on the current [VideoCompressor](#) and [RecordingMethod](#), so be sure to first select the RecordingMethod() and VideoCompressor before saving its settings.

The **IsVideoCompressor** parameter must be set to **true** for a video compressor, or **false** for an audio compressor.

The Compressor parameter index must be the [VideoCompressor](#) or [AudioCompressor](#) index of the related compressor in the [VideoCompressors](#) or [AudioCompressors](#) lists.

See [Saving and restoring compressor settings programmatically](#) for sample code.

Created with the Standard Edition of HelpNDoc: [Elevate your documentation to new heights with HelpNDoc's built-in SEO](#)

## LoadCompressorSettingsFromTextFile

**TVideoGrabber.LoadCompressorSettingsFromTextFile**

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Reloads the compressor settings from a text file.

**Declaration**

**function** LoadCompressorSettingsFromTextFile (IsVideoCompressor: boolean; CompressorIndex: LongInt; FileName: **string**): Boolean;

**bool** \_\_fastcall LoadCompressorSettingsFromTextFile (**bool** IsVideoCompressor, **int** CompressorIndex: LongInt, wchar\_t \*FileName);

function LoadCompressorSettingsFromTextFile (IsVideoCompressor as Boolean, CompressorIndex as Long, FileName as String) As String

**Description**

Used to reload the compressor settings from a text file.

The compressor settings depend on the current [VideoCompressor](#) and [RecordingMethod](#), so be sure to first select the RecordingMethod() and VideoCompressor before saving its settings.

The **IsVideoCompressor** parameter must be set to **true** for a video compressor, or **false** for an audio compressor.

The Compressor parameter index must be the [VideoCompressor](#) or [AudioCompressor](#) index of the related compressor in the [VideoCompressors](#) or [AudioCompressors](#) lists.

See [Saving and restoring compressor settings programmatically](#) for sample code.

---

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

---

**MixAudioSamples****TVideoGrabber.MixAudioSamples**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Mixes the audio samples passed as parameter

**Declaration**

**function** MixAudioSamples (pSampleBuffer: Pointer; SampleBufferSize: LongInt; SampleDataLength: LongInt; SampleFormatType: TFormatType; pFormat: Pointer; SampleStartTime: LargeInteger; SampleStopTime: LargeInteger): Boolean;

**bool** \_\_fastcall MixAudioSamples(void \*pSampleBuffer, int SampleBufferSize, int SampleDataLength, TFormatType SampleFormatType, void \*pFormat, \_\_int64 SampleStartTime, \_\_int64 SampleStopTime);

function \_\_fastcall MixAudioSamples (pSampleBuffer as Long, SampleBufferSize as Long, SampleDataLength as Long, SampleFormatType as TFormatType, pFormat as Long, SampleStartTime as Double, SampleStopTime as Double) as Bool

**Description**

Used to mix the audio samples coming out from another TVideoGrabber component.

**See Also**

[MixAudioSamples\\_CurrentSourceLevel](#) [MixAudioSamples\\_ExternalSourceLevel](#)  
[Mixer\\_SetupPIPFromSource](#)

---

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

**Mixer\_Activation****TVideoGrabber.Mixer\_Activation**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables / disables temporarily a mixing

#### Declaration

**function** Mixer\_Activation(Id: LongInt; Activate: Boolean): Boolean;

**bool \_\_fastcall** Mixer\_Activation(**int** Id, **bool** Activate)

Function Mixer\_Activation (Id as Long, Activate as Boolean) as Boolean

#### Description

Used to enable / disable the video mixing of a source component in the mixer component.

Pass as parameter the mixer Id returned by [Mixer\\_AddToMixer](#) when associating the source component to the mixer.

E.g. VideoGrabber1 previews a video capture device, and VideoGrabber2 "duplicates" the video frames from VideoGrabber1, like it was a 2nd component capturing the same device:

- to start the preview of the capture device:

```
VideoGrabber1.VideoSource = vs_VideoCaptureDevice
VideoGrabber1.StartPreview
```

- to start the 2nd component that will preview the frames of the 1st component, in mixer mode:

```
VideoGrabber2.VideoSource = vs_Mixer
MixerId = VideoGrabber2.Mixer_AddToMixer (VideoGrabber1.UniqueID, 0, 0, 0, 0, 0, true, true)
VideoGrabber2.StartPreview
```

- to disable temporarily the mixing:

```
VideoGrabber2.Mixer_Activation (MixerId, false)
```

- to resume temporarily the mixing:

```
VideoGrabber2.Mixer_Activation (MixerId, true)
```

#### See Also

[Mixer\\_AddToMixer](#) [Mixer\\_MosaicColumns](#) [Mixer\\_MosaicLines](#) [Mixer\\_RemoveFromMixer](#)  
[Mixer\\_SetOverlayRoundedCorner](#) [Mixer\\_SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

## Mixer\_AddToMixer

### TVideoGrabber.Mixer\_AddToMixer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to associate 2 TVideoGrabber components in [Mixer](#) mode

#### Declaration

**function** Mixer\_AddToMixer (SourceUniqueID: LongInt; SourceVideoInput: LongInt; MosaicLine: LongInt; MosaicColumn: LongInt; AlternatedGroup: LongInt; AlternatedTimeIntervalInMs: LongInt; ReplacePreviouslyAdded: Boolean; EraseBackground: Boolean): LongInt;

**int \_\_fastcall** Mixer\_AddToMixer(**int** SourceUniqueID, **int** SourceVideoInput, **int** MosaicLine, **int** MosaicColumn, **int** AlternatedGroup, **int** AlternatedTimeIntervalInMs, **bool** ReplacePreviouslyAdded, **bool** EraseBackground);

function Mixer\_AddToMixer (SourceUniqueID as Long, SourceVideoInput as Long, MosaicLine as Long,

MosaicColumn as Long, AlternatedGroup as Long, AlternatedTimeIntervalInMs as Long, ReplacePreviouslyAdded as Bool EraseBackground as Bool) as long

### Description

This function is invoked **on the [mixer](#) component** to associate a source TVideoGrabber component in [Mixer](#) mode.

(the mixer component must have been set in this mode by setting [VideoSource](#) = vs\_Mixer).

You will find sample code of this function in the "[How to mix several video sources into one a single one](#)" chapter.

### Parameters:

#### SourceUniqueID:

UniqueID of the source component

#### SourceVideoInput:

*not used yet, set it to 0*

#### MosaicLine:

Line where the source video must be placed when displaying several sources on a mosaic layout (set to 0 if the mosaic layout is not used).

#### MosaicColumn:

Column where the source video must be placed when displaying several sources on a mosaic layout (set to 0 if the mosaic layout is not used).

#### AlternatedGroup:

Group in which the source video will be displayed alternatively.

You can specify any number.

The rule is: all sources associated to this mixer with this group number will be displayed at the same time, and never displayed at the same time than other sources associated to this mixer with other group numbers.

#### AlternatedTimeIntervalInMs:

Duration of display of the sources having this group number in the mixer.

The duration is expressed in milliseconds.

*Remark: if several sources are associated to a mixer with the same group number but not the same duration, only one of these durations will be taken in account when this group will be displayed.*

*Therefore to prevent confusion, each sources of a given group should logically be associated to a mixer with the same duration.*

#### ReplacePreviouslyAdded:

Specifies if invoking Mixer\_AddToMixer on a given video source replaces any previous Mixer\_AddToMixer setting already set. **Recommended value: TRUE**

#### EraseBackground:

specifies if the background must be erased on the mixer when Mixer\_AddToMixer is invoked while the mixer is running. **Recommended value: TRUE**

#### RETURN VALUE:

The return value of this function is an identifier that may be used to remove the current Mixer\_AddToMixer setting by invoking [Mixer\\_RemoveFromMixer](#).

### See Also

[Mixer\\_Activation](#) [Mixer\\_MosaicColumns](#) [Mixer\\_MosaicLines](#) [Mixer\\_RemoveFromMixer](#)  
[Mixer\\_SetOverlayRoundedCorner](#) [Mixer\\_SetupPIPFromSource](#)

## Mixer\_RemoveFromMixer

### TVideoGrabber.Mixer\_RemoveFromMixer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to remove the association of 2 TVideoGrabber components in [Mixer](#) mode

#### Declaration

**function** Mixer\_RemoveFromMixer (Id: LongInt): Boolean;

**bool** \_\_fastcall Mixer\_RemoveFromMixer(**int** Id);

function Mixer\_RemoveFromMixer (Id as Long) as Bool

#### Description

Invoke Mixer\_RemoveFromMixer to remove an between a source component and a Mixer component previously established by invoking [Mixer\\_AddToMixer](#).

Pass as parameter the identifier that was returned by the [Mixer\\_AddToMixer](#) function when it has been invoked.

#### See Also

[Mixer\\_Activation](#) [Mixer\\_AddToMixer](#) [Mixer\\_MosaicColumns](#) [Mixer\\_MosaicLines](#) [Mixer\\_SetOverlayRoundedCorner](#) [Mixer\\_SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## Mixer\_SetOverlayRoundedCorner

### TVideoGrabber.Mixer\_SetOverlayRoundedCorner

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Rounds the corners of an image overlay

#### Declaration

**function** Mixer\_SetOverlayRoundedCorner (Id: LongInt; CornerRadiusWidth: LongInt; CornerRadiusHeight: LongInt): Boolean;

**bool** \_\_fastcall Mixer\_SetOverlayRoundedCorner(**int** Id, **int** CornerRadiusWidth, **int** CornerRadiusHeight);

#### Description

Used to make an image overlay appear in the mixer with rounded corners or elipsoïdal, depending on the values specified

#### See Also

[Mixer\\_Activation](#) [Mixer\\_AddToMixer](#) [Mixer\\_MosaicColumns](#) [Mixer\\_MosaicLines](#) [Mixer\\_RemoveFromMixer](#) [Mixer\\_SetupPIPFromSource](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)

## Mixer\_SetupPIPFromSource

### TVideoGrabber.Mixer\_SetupPIPFromSource

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to make a PIP (Picture In Picture) between 2 TVideoGrabber components

## Declaration

**function** Mixer\_SetupPIPFromSource (SourceUniqueID: LongInt; Source\_Left: LongInt; Source\_Top: LongInt; Source\_Width: LongInt; Source\_Height: LongInt; ActivatePIP: Boolean; PIP\_Left: LongInt; PIP\_Top: LongInt; PIP\_Width: LongInt; PIP\_Height: LongInt; MoveToTop: Boolean): LongInt;

**int \_\_fastcall** Mixer\_SetupPIPFromSource(**int** SourceUniqueID, **int** Source\_Left, **int** Source\_Top, **int** Source\_Width, **int** Source\_Height, **bool** ActivatePIP, **int** PIP\_Left, **int** PIP\_Top, **int** PIP\_Width, **int** PIP\_Height, **bool** MoveToTop);

function SetPIPFromSource (SourceUniqueID as Long, Source\_Left as Long, Source\_Top as Long, Source\_Width as Long, Source\_Height as Long, ActivatePIP as Bool, PIP\_Left as Long, PIP\_Top as Long, PIP\_Width as Long, PIP\_Height as Long, MoveToTop as Bool) as Long

## Description

This function setups a TVideoGrabber component to receive the whole video frames (or a cropped part of these video frames) from another TVideoGrabber component, and to display it in "PIP" mode (Picture In Picture), at any location or size in the destination video frame.

Parameters:

- the "Source\_..." parameters specify a "rectangle part" of the source video frame that will be displayed in the destination video frame. Specifying 0, 0, 0, 0 means you send the whole source video frame.
- the "PIP\_..." parameters specify a "rectangle part" of the destination video frame where the source will be displayed in "PIP" mode.

### SourceUniqueID: LongInt

specifies the UniqueID property of the TVideoGrabber component that must send the video frames

### Source\_Left: LongInt

Left location in the source video frame

### Source\_Top: LongInt

Top location in the source video frame

### Source\_Width: LongInt

Width of the source rectangle (set 0 to let TVideoGrabber use automatically the full width of the source video frame)

### Source\_Height: LongInt

Height of the source rectangle (set 0 to let TVideoGrabber use automatically the full width of the source video frame)

### ActivatePIP: Boolean

Specifies if the PIP display must be activated or not

### PIP\_Left: LongInt

Left location of the PIP on the destination video frame

### PIP\_Top: LongInt

Top location of the PIP on the destination video frame

### PIP\_Width: LongInt

Width of the PIP on the destination video frame

### PIP\_Height: LongInt

Height of the PIP on the destination video frame

### MoveToTop: Boolean

Must be set to true if several PIP are partially overlapped and this one should appear above the others

### RETURN VALUE: LongInt



Identifier of the mixer relation, this value is required e.g. by [Mixer\\_Activation](#) to activate/deactivate this PIP on the fly.

#### See Also

[MixAudioSamples](#) [MixAudioSamples\\_CurrentSourceLevel](#) [MixAudioSamples\\_ExternalSourceLevel](#) [Mixer\\_Activation](#) [Mixer\\_AddToMixer](#) [Mixer\\_MosaicColumns](#) [Mixer\\_MosaicLines](#) [Mixer\\_RemoveFromMixer](#) [Mixer\\_SetOverlayRoundedCorner](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

## Monitor\_Primary\_Index

### TVideoGrabber.Monitor\_Primary\_Index

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

returns the index of the primary monitor

#### Declaration

**function** Monitor\_Primary\_Index: LongInt;

**int** \_\_fastcall Monitor\_Primary\_Index();

function Monitor\_Primary\_Index() As Long

#### Description

Used to retrieve the index of the primary monitor, in the 0..n-1 range

#### See Also

[Dual display](#) [TVideoRenderer\\_AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_SetLocation](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

## MonitorBounds

### TVideoGrabber.MonitorBounds

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the bounds of the specified video screen, in pixels.

#### Declaration

**function** MonitorBounds(MonitorNumber: LongInt; **var** Bounds: TRect): Boolean;

**bool** \_\_fastcall MonitorBounds(**int** MonitorNumber, Windows::TRect &Bounds);

Function MonitorBounds(MonitorNumber As Long, LeftBound, TopBound, RightBound, BottomBound) As Boolean

### Description

Returns the bounds of the specified video screen, in pixels.

Note: the MonitorNumber parameter is in the [ 0 .. [MonitorsCount](#) - 1 ] range.

Useful on platforms that have more than one video screen.

See the [Dual display](#) chapter for more information.

### See Also

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#) [DualDisplay](#) [Left](#) [DualDisplay](#) [Monitor](#) [DualDisplay](#) [MouseMovesWindow](#) [DualDisplay](#) [PanScanRatio](#) [DualDisplay](#) [SetLocation](#) [DualDisplay](#) [StayOnTop](#) [DualDisplay](#) [Top](#) [DualDisplay](#) [TransparentColorEnabled](#) [DualDisplay](#) [TransparentColorValue](#) [DualDisplay](#) [VideoHeight](#) [DualDisplay](#) [VideoPortEnabled](#) [DualDisplay](#) [VideoWidth](#) [DualDisplay](#) [VideoWindowHandle](#) [DualDisplay](#) [Visible](#) [DualDisplay](#) [Width](#) [IsVideoPortAvailable](#) [Monitor](#) [Primary](#) [Index](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer

## MonitorsCount

### TVideoGrabber.MonitorsCount

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Number of monitors available on the current platform.

### Declaration

function MonitorsCount: LongInt;

int \_\_fastcall MonitorsCount(void);

Function MonitorsCount() As Long

### Description

Returns the number of monitors available on the current platform.

Useful on platforms that have more than one video screen.

See the [Dual display](#) chapter for more information.

### See Also

[Dual display](#) [TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display](#) [Active](#) [Display](#) [AlphaBlendEnabled](#) [Display](#) [AlphaBlendValue](#) [Display](#) [AutoSize](#) [Display](#) [Embedded](#) [Display](#) [FullScreen](#) [Display](#) [Height](#) [Display](#) [Left](#) [Display](#) [Monitor](#) [Display](#) [MouseMovesWindow](#) [Display](#) [PanScanRatio](#) [Display](#) [SetLocation](#) [Display](#) [StayOnTop](#) [Display](#) [Top](#) [Display](#) [TransparentColorEnabled](#) [Display](#) [TransparentColorValue](#) [Display](#) [VideoHeight](#) [Display](#) [VideoPortEnabled](#) [Display](#) [VideoWidth](#) [Display](#) [VideoWindowHandle](#) [Display](#) [Width](#) [DualDisplay](#) [Active](#) [DualDisplay](#) [AlphaBlendEnabled](#) [DualDisplay](#) [AlphaBlendValue](#) [DualDisplay](#) [AutoSize](#) [DualDisplay](#) [Embedded](#) [DualDisplay](#) [FullScreen](#) [DualDisplay](#) [Height](#)

[DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_SetLocation](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

## MotionDetector\_CellColorIntensity

### TVideoGrabber.MotionDetector\_CellColorIntensity

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the color intensity of a cell in the grid.

#### Declaration

**function** MotionDetector\_CellColorIntensity (RGBSelector: TRGBSelector; x, y: LongInt): LongInt;

**int** \_\_fastcall MotionDetector\_CellColorIntensity(TRGBSelector RGBSelector, **int** x, **int** y);

function MotionDetector\_CellColorIntensity (RGBSelector as TRGBSelector, x as Long, y as Long) as Long

#### Description

Retrieves the current average color intensity of a cell in the grid.

The [RGBSelector](#) parameter specifies the color to retrieve (R, G or B).

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## MotionDetector\_CellMotionRatio

### TVideoGrabber.MotionDetector\_CellMotionRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the motion ratio of a cell on the grid.

#### Declaration

**function** MotionDetector\_CellMotionRatio(x, y: LongInt): Double;

**double** \_\_fastcall MotionDetector\_CellMotionRatio(**int** x, **int** y)

Function MotionDetector\_CellMotionRatio(x As Long, y As Long) As Double

#### Description

Used to retrieve the motion ratio (cell's moved pixels count / cell's total pixels count) of the cell specified by its x and y coordinates. The return value will be in the 0...1 range.

The x and y coordinates are zero-based index, from 0 to [MotionDetector\\_GridXCount](#) -1 and 0 to [MotionDetector\\_GridYCount](#) -1.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#)

[MotionDetector\\_Grid](#)
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Quickly and Easily Convert Your Word Document to an ePub or Kindle eBook](#)

## MotionDetector\_EnumGridDialogControls

### TVideoGrabber.MotionDetector\_EnumGridDialogControls

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Methods](#)

Gives access to the controls of the grid dialog.

#### Declaration

**function** MotionDetector\_EnumGridDialogControls(FirstControl: Boolean): TComponent;

Classes::TComponent \*\_\_fastcall MotionDetector\_EnumGridDialogControls(**bool** FirstControl)

n/a

#### Description

This function enumerates the the controls of the grid dialog, including the dialog form.

This lets you e.g. customize the button names, the dialog size, etc...

- invoking MotionDetector\_EnumGridDialogControls(True) returns the first control (in fact the dialog form), and resets the counter,
- invoking MotionDetector\_EnumGridDialogControls(False) returns the next control, if any, or **nil** when the last control of the dialog has been reached.

E.g.:

```

procedure TForm1.Button1Click(Sender: TObject);
var
    Control: TComponent;
begin
    VideoGrabber1.MotionDetector_ShowGridDialog; // BEFORE...

    Control := VideoGrabber1.MotionDetector_EnumGridDialogControls (True); // reset the counter
    while assigned (Control) do begin // for each control...
        Memo1.Lines.Add (Control.Name + ': ' + Control.ClassName); // shows the control name
        if Control.Name = 'frmGridDialog' then begin
            TForm(Control).Caption := 'this is my custom dialog title';
        end
        else if Control.Name = 'btnOK' then begin
            TButton(Control).Caption := 'DONE';
        end
        else if Control.Name = 'btnCancel' then begin
            TButton(Control).Caption := 'Discard';
        end;
        Control := VideoGrabber1.MotionDetector_EnumGridDialogControls (False); // returns the next control
    end;

    VideoGrabber1.MotionDetector_ShowGridDialog; // AFTER.

```

```
end;
```

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#)
[Motion ratio](#)
[Recording only when motion is detected](#)  
[Video noise](#)
[TOnMotionDetected](#)
[TOnMotionNotDetected](#)
[MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#)
[MotionDetector\\_CompareGreen](#)
[MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#)
[MotionDetector\\_Get2DTextGrid](#)
[MotionDetector\\_Get2DTextMotion](#)  
[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)
[MotionDetector\\_GetCellSize](#)  
[MotionDetector\\_GloballyIncOrDecSensitivity](#)
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)  
[MotionDetector\\_Grid](#)
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)  
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)  
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Free EPub and documentation generator

## MotionDetector\_Get2DTextGrid

### TVideoGrabber.MotionDetector\_Get2DTextGrid

[Prev](#)
[Next](#)

[TVideoGrabber](#)   [Methods](#)  
[ber](#)

Retrieves a pseudo 2D text representation of the grid.

#### Declaration

**function** MotionDetector\_Get2DTextGrid: **string**;

wchar\_t \* \_\_fastcall MotionDetector\_Get2DTextGrid()

Function MotionDetector\_Get2DTextGrid() As String

#### Description

Used to retrieve a pseudo 2D text representation of the grid. Useful for debug / finalization purpose.

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#)
[Motion ratio](#)
[Recording only when motion is detected](#)  
[Video noise](#)
[TOnMotionDetected](#)
[TOnMotionNotDetected](#)
[MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#)
[MotionDetector\\_CompareGreen](#)
[MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#)
[MotionDetector\\_EnumGridDialogControls](#)
[MotionDetector\\_Get2DTextMotion](#)  
[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)
[MotionDetector\\_GetCellSize](#)  
[MotionDetector\\_GloballyIncOrDecSensitivity](#)
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)  
[MotionDetector\\_Grid](#)
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered](#)
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)  
[OnMotionDetected](#)
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)  
[RecordingOnMotion\\_MotionThreshold](#)
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Eliminate the Struggles of Documentation with a Help Authoring Tool

## MotionDetector\_Get2DTextMotion

### TVideoGrabber.MotionDetector\_Get2DTextMotion

[Prev](#)
[Next](#)

## **TVideoGrabber** [Methods](#)

Retrieves a pseudo 2D text representation of the motion.

### **Declaration**

**function** MotionDetector\_Get2DTextMotion: **string**;

wchar\_t \* \_\_fastcall MotionDetector\_Get2DTextMotion()

Function MotionDetector\_Get2DTextMotion() As String

### **Description**

Used to retrieve a pseudo 2D text representation of the motion. Useful for debug / finalization purpose.

### **See Also**

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion](#) [MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Bring your WinHelp HLP help files into the present with HelpNDoc's easy CHM conversion](#)

## **MotionDetector\_GetCellLocation**

### **TVideoGrabber.MotionDetector\_GetCellLocation**

[Prev](#)

[Next](#)

## **TVideoGrabber** [Methods](#)

Retrieves the x an y coordinates of a cell in the grid.

### **Declaration**

**function** MotionDetector\_GetCellLocation(x, y: LongInt; **var** XLocation, YLocation: LongInt): Boolean;

**bool** \_\_fastcall MotionDetector\_GetCellLocation(**int** x, **int** y, **int** &XLocation, **int** &YLocation)

Function MotionDetector\_GetCellLocation(x As Long, y As Long, X\_Location, Y\_Location) As Boolean

### **Description**

Used to retrieve the x an y coordinates of a cell in the grid.

### **See Also**

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#)



[MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## MotionDetector\_GetCellSensitivity

### TVideoGrabber.MotionDetector\_GetCellSensitivity

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the sensitivity of a cell in the grid.

#### Declaration

**function** MotionDetector\_GetCellSensitivity(x, y: LongInt; **var** Value: LongInt): Boolean;

**bool** \_\_fastcall MotionDetector\_GetCellSensitivity(int x, int y, int &Value)

Function MotionDetector\_GetCellSensitivity(x As Long, y As Long, Value as Long)As Boolean

#### Description

Used to retrieve the sensitivity of a cell in the grid.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

## MotionDetector\_GetCellSize

### TVideoGrabber.MotionDetector\_GetCellSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the x and y sizes of a cell in the grid.

#### Declaration

**function** MotionDetector\_GetCellSize(**var** XSize, YSize: LongInt): Boolean;

**bool** \_\_fastcall MotionDetector\_GetCellSize(int &XSize, int &YSize)

Function MotionDetector\_GetCellSize(X\_Size, Y\_Size) As Boolean

#### Description

Used to retrieve the x and y sizes of a cell in the grid.



## See Also

[Color / Greyscale](#) [Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#) [MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#) [MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#) [MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#) [MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#) [MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#) [MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#) [MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

## MotionDetector\_GlobalColorIntensity

### TVideoGrabber.MotionDetector\_GlobalColorIntensity

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the global average color intensity of the last video frame.

#### Declaration

**function** MotionDetector\_GlobalColorIntensity (RGBSelector: TRGBSelector): LongInt;

**int** \_\_fastcall MotionDetector\_GlobalColorIntensity(TRGBSelector RGBSelector);

**function** MotionDetector\_GlobalColorIntensity (RGBSelector as TRGBSelector) as Long

#### Description

Used to retrieve the global average color intensity of the last video frame.  
The [RGBSelector](#) parameter specifies the color to retrieve (R, G or B).

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## MotionDetector\_GloballyIncOrDecSensitivity

### TVideoGrabber.MotionDetector\_GloballyIncOrDecSensitivity

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Increases or decreases the sensitivity of the grid cells.

#### Declaration

**procedure** MotionDetector\_GloballyIncOrDecSensitivity(Value: integer);

**void** \_\_fastcall MotionDetector\_GloballyIncOrDecSensitivity(int Value)

Sub MotionDetector\_GloballyIncOrDecSensitivity(Value As Long)

#### Description

Used to globally increase or decrease the sensitivity of the cells whose current sensitivity is greater than 0 (this increases or decreases only the sensitivity of the active cells).  
The value passed as parameter is usually -1 (decrease) or 1 (increase).

**See Also**

[Color / Greyscale](#) [Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#)  
[Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize](#) [MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#)  
[MotionDetector\\_Grid](#) [MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#)  
[OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#)  
[RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Elevate Your CHM Help Files with HelpNDoc's Advanced Customization Options

**MotionDetector\_Reset****TVideoGrabber.MotionDetector\_Reset**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Disables temporarily the motion detection for the next frame.

**Declaration**

**procedure** MotionDetector\_Reset;

**void** \_\_fastcall MotionDetector\_Reset(**void**)

Sub MotionDetector\_Reset()

**Description**

Used to disable temporarily the motion detection for the next frame.

Call this method if the next frame is not frame that normally follows the previous one currently stored by the component (e.g. the motion detection has been stopped for a moment, now it restarts but you don't want the 1st frame to be interpreted as motion because it differs from the last one stored a few minutes ago).

**See Also**

[Color / Greyscale](#) [Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#)  
[Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#)  
[MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#)  
[OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#)  
[RecordingOnMotion\\_MotionThreshold](#) [RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Full-featured Documentation generator

**MotionDetector\_ResetGlobalSensitivity****TVideoGrabber.MotionDetector\_ResetGlobalSensitivity**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

## ber

Gobally sets/resets sensitivity of all cells.

### Declaration

**procedure** MotionDetector\_ResetGlobalSensitivity(Value: integer);

**void** \_\_fastcall MotionDetector\_ResetGlobalSensitivity(**int** Value)

Sub MotionDetector\_ResetGlobalSensitivity(Value As Long)

### Description

Used to globally set/reset the sensitivity of all cells, in the 0..9 range.

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_ShowGridDialog MotionDetector\\_Triggered MotionDetector\\_UseThisReferenceSample OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected RecordingOnMotion Enabled RecordingOnMotion MotionThreshold RecordingOnMotion NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## MotionDetector\_SetCellSensitivity

### TVideoGrabber.MotionDetector\_SetCellSensitivity

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the sensitivity of one cell in the grid.

### Declaration

**function** MotionDetector\_SetCellSensitivity(x, y, Value: LongInt): Boolean;

**bool** \_\_fastcall MotionDetector\_SetCellSensitivity(**int** x, **int** y, **int** Value)

Function MotionDetector\_SetCellSensitivity(x As Long, y As Long, Value As Long) As Boolean

### Description

Used to set the sensitivity of one cell in the grid.

Useful if the dialog cannot be used to define sensitivity areas because the grid is too dense (usually above 40 x 40).

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid](#)

[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetGridSize](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)  
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)  
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

## MotionDetector\_SetGridSize

### TVideoGrabber.MotionDetector\_SetGridSize

[Prev](#)

[Next](#)

[TVideoGrabber](#)  
[Methods](#)

Modifies the size of the grid.

#### Declaration

**procedure** MotionDetector\_SetGridSize(x, y: LongInt);

**void** \_\_fastcall MotionDetector\_SetGridSize(int x, int y)

Sub MotionDetector\_SetGridSize(x As Long, y As Long)

#### Description

Used to modify the size of the grid.

The grid can be set from 1 x 1 to the *width x height* of the video frame.

If the grid oversize the frame, only the top/left part of the grid that covers the frame is active.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#)
[Motion ratio](#)
[Recording only when motion is detected](#)  
[Video noise](#)
[TOnMotionDetected](#)
[TOnMotionNotDetected](#)
[MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#)
[MotionDetector\\_CompareGreen](#)
[MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#)
[MotionDetector\\_EnumGridDialogControls](#)
[MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion](#)
[MotionDetector\\_GetCellLocation](#)
[MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize](#)
[MotionDetector\\_GloballyIncOrDecSensitivity](#)  
[MotionDetector\\_GlobalMotionRatio](#)
[MotionDetector\\_GreyScale](#)
[MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount](#)
[MotionDetector\\_GridYCount](#)
[MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#)
[MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#)
[MotionDetector\\_Reset](#)
[MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#)
[MotionDetector\\_ShowGridDialog](#)
[MotionDetector\\_Triggered](#)  
[MotionDetector\\_UseThisReferenceSample](#)
[OnBacktimedFramesCountReached](#)
[OnMotionDetected](#)  
[OnMotionNotDetected](#)
[RecordingOnMotion\\_Enabled](#)
[RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

## MotionDetector\_ShowGridDialog

### TVideoGrabber.MotionDetector\_ShowGridDialog

[Prev](#)

[Next](#)

[TVideoGrabber](#)  
[Methods](#)

Displays the grid dialog.

#### Declaration

**procedure** MotionDetector\_ShowGridDialog;

**void** \_\_fastcall MotionDetector\_ShowGridDialog(void)

Sub MotionDetector\_ShowGridDialog()

### Description

Used to display the grid dialog that allows to set the sensitivity of each cell interactively. If at least one frame has been processed, the dialog shows the grid over the video frame, allowing to adjust accurately the sensitivity of the video areas.

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected](#)  
[Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity](#)  
[MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_Triggered](#)  
[MotionDetector\\_UseThisReferenceSample OnBacktimedFramesCountReached OnMotionDetected](#)  
[OnMotionNotDetected RecordingOnMotion\\_Enabled RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## MotionDetector\_TriggerNow

### TVideoGrabber.MotionDetector\_TriggerNow

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Triggers a motion detection.

### Declaration

**procedure** MotionDetector\_TriggerNow;

**void** \_\_fastcall MotionDetector\_TriggerNow(**void**)

Sub MotionDetector\_TriggerNow()

### Description

Used to trigger a motion detection, when the [MotionDetector\\_Triggered](#) property is enabled. After invoking this method, the motion detection will occur one time for the next video frame, until the next MotionDetector\_TriggerNow call.

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## MotionDetector\_UseThisReferenceSample

### TVideoGrabber.MotionDetector\_UseThisReferenceSample

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Let specify a reference sample used for motion detection.

### Declaration

**function** MotionDetector\_UseThisReferenceSample (Bitmap: TBitmap; BMPFile: **string**; JPEGFile: **string**): **Boolean**;

**bool** \_\_fastcall MotionDetector\_UseThisReferenceSample (Graphics::TBitmap \*TBitmap, wchar\_t \*BMPFile, wchar\_t \*JPEGFile)

Function MotionDetector\_UseThisReferenceSample (Bitmap as Long, BMPFile as string, JPEGFile as string) As Boolean

### Description

Let specify a video frame that will be used as reference for motion detection, instead of the previous frame.

The frame can be passed in 3 ways, by using one of the 3 parameters:

1. **Bitmap:** as a pointer to a TBitmap (or a bitmap handle in the OCX versions), e.g.:  
VideoGrabber.MotionDetector\_UseThisReferenceSample (MyTBitmapReference, "", "");
2. **BMPFile:** as a path to a BMP image, e.g.:  
VideoGrabber.MotionDetector\_UseThisReferenceSample (nil, "MyBitmapReference.bmp", "");
3. **JPEGFile:** as a path to a JPEG file, e.g.:  
VideoGrabber.MotionDetector\_UseThisReferenceSample (nil, "", "MyBitmapReference.jpg");

To deactivate the feature and reuse the last frame as reference, pass empty parameters, e.g.:  
VideoGrabber.MotionDetector\_UseThisReferenceSample (nil, "", "");

### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected](#)  
[Video noise TOnMotionDetected TOnMotionNotDetected MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue MotionDetector\\_CompareGreen MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled MotionDetector\\_EnumGridDialogControls MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion MotionDetector\\_GetCellLocation MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize MotionDetector\\_GloballyIncOrDecSensitivity](#)  
[MotionDetector\\_GlobalMotionRatio MotionDetector\\_GreyScale MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount MotionDetector\\_GridYCount MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise MotionDetector\\_Reset MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity MotionDetector\\_SetGridSize MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected](#)  
[RecordingOnMotion\\_Enabled RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: Make Documentation Review a Breeze with HelpNDoc's  
 Advanced Project Analyzer

## MP4NeedsReindexing

### TVideoGrabber.MP4NeedsReindexing

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Determines if a MP4 video clip requires reindexing.

### Declaration

**function** TVideoGrabber.MP4NeedsReindexing (FilePath: **string**): TTriState;

TTriState \_\_fastcall MP4NeedsReindexing(System::UnicodeString FilePath);

### Description

Used to determines if a MP4 video clip requires a reindexing.

Return values:

ts\_Undefined: failed to open the clip

ts\_False: the clip does not need a reindexing



ts\_True: the clip needs to be reindexed

See [MultipurposeEncoder\\_ReindexClip](#)

#### See Also

[Player](#) [features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [TVideoDeinterlacing](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MultipurposeEncoder\\_ReindexClip](#) [OnFrameBitmap](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchrone](#) [PlayerVideoCodec](#) [RetrievalInitialXYAfterRotation](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoProcessing](#) [Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [VideoWidth](#) [PreferredAspectRatio](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation

## MPEGProgramSetting

### TVideoGrabber.MPEGProgramSetting

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the program setting to be used

#### Declaration

**function** MPEGProgramSetting (MPEGProgramSetting: TMPEGProgramSetting; Value: LongInt): LongInt;

**int \_\_fastcall** MPEGProgramSetting(TMPEGProgramSetting MPEGProgramSetting, **int** Value);

**function** MPEGProgramSetting (MPEGProgramSetting as TxMPEGProgramSetting, Value as Long): as Long

#### Description

Used to specify the [TMPEGProgramSetting](#) MPEG program number to be used when an UDP MPEG Transport Stream network source is played

Created with the Standard Edition of HelpNDoc: Easily create Qt Help files

## MultipurposeEncoder\_QuickConfigure\_UDPStreaming\_H264

### TVideoGrabber.MultipurposeEncoder\_QuickConfigure\_UDPStreaming\_H264

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Quick configure the Multipurpose Encoder for H264 streaming (unicast or multicast)

#### Declaration

**function** MultipurposeEncoder\_QuickConfigure\_UDPStreaming\_H264 (LogToFile: Boolean; VideoEnabled: Boolean; AudioEnabled: Boolean; DestinationIP: **string**; DestinationPort: LongInt; VideoBitRateKb: LongInt; AudioBitRateKb: LongInt): Boolean;



System::wchar\_t \* \_\_fastcall MultipurposeEncoder\_QuickConfigure\_UDPStreaming\_H264(**bool** LogToFile, **bool** VideoEnabled, **bool** AudioEnabled, System::wchar\_t \*DestinationIP, **int** DestinationPort, **int** VideoBitRateKb, **int** AudioBitRateKb);

### Description

Parameters:

**LogToFile**: Boolean: if true, a log file is generated in the current folder

**VideoEnabled**: Boolean: if true, the video is streamed

**AudioEnabled**: Boolean: if true, the audio is streamed

**DestinationIP**: string: destination IP (unicast or multicast)

**DestinationPort**: LongInt: destination port

**VideoBitRateKb**: LongInt: the video bitrate expressed in Kb/s, e.g. 2000Kbs

**AudioBitRateKb**: LongInt: the audio bitrate expressed in Kb/s, e.g. 128Kbs

### See Also

[MultipurposeEncoder\\_ReindexClip](#) [SetFFmpegAudioFilter](#) [SetFFmpegFilter](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

## MultipurposeEncoder\_ReindexClip

### TVideoGrabber.MultipurposeEncoder\_ReindexClip

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to reindex a video clip

### Declaration

**function** MultipurposeEncoder\_ReindexClip (CurrentFileName: **string**; NewFileName: **string**): Boolean;

**bool** \_\_fastcall MultipurposeEncoder\_ReindexClip(System::UnicodeString CurrentFileName, System::UnicodeString NewFileName);

### Description

This function reindexes the video clip specified as **CurrentFileName** video clip and produces a new video clip specified by **NewFileName**.

The reindexing is performed without recompressing the video stream, so the process is quite fast.

The function is not specific to MP4, it can reindex MKV, AVI or other formats.

It is useful when seeking a video clip is slow.

You can determine if a MP4 video clip needs a reindexing by invoking [MP4NeedsReindexing](#).

E.g.:

SourceFileName = "c:

NewFileName := "c:

if (VideoGrabber->MP4NeedsReindexing (SourceFileName))

VideoGrabber->MultipurposeEncoder\_ReindexClip (SourceFileName, NewFileName);

### See Also

[MP4NeedsReindexing](#) [MultipurposeEncoder\\_QuickConfigure\\_UDPStreaming\\_H264](#) [SetFFmpegAudioFilter](#) [SetFFmpegFilter](#)

Created with the Standard Edition of HelpNDoc: [News and information about help authoring tools and software](#)

## NotifyPlayerTrackbarAction

### TVideoGrabber.NotifyPlayerTrackbarAction

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Invoked by your trackbar events

#### Declaration

**procedure** NotifyPlayerTrackbarAction (TrackbarAction: TTrackbarAction);

**void \_\_fastcall** NotifyPlayerTrackbarAction(TTrackbarAction TrackbarAction);

Sub NotifyPlayerTrackbarActio (TrackbarAction as TxTrackbarAction)

#### Description

If you are implementing a player trackbar, invoke this procedure from your trackbar's OnMouseDown, OnMouseUp, OnKeyDown and OnKeyUp events.

See the player's [Trackbar](#) chapter for more information.

Created with the Standard Edition of HelpNDoc: [Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

## ONVIF\_GetStr

### TVideoGrabber.ONVIF\_GetStr

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves settings from the Datstead [RTSP/RTMP/HTTP/ONVIF DirectShow Source Filter](#)

#### Declaration

**function** ONVIF\_GetStr (ParamIdentifier: **string**; var Value: **string**): Boolean;

**bool \_\_fastcall** ONVIF\_GetStr(System::wchar\_t \*ParamIdentifier, System::wchar\_t \*&Value);

#### Description

Used to retrieve settings from the Datstead [RTSP/RTMP/HTTP/ONVIF DirectShow Source Filter](#)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## ONVIF\_SetStr

### TVideoGrabber.ONVIF\_SetStr

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Pass settings to the Datstead [RTSP/RTMP/HTTP/ONVIF DirectShow Source Filter](#)

#### Declaration

**function** ONVIF\_SetStr (ParamIdentifier: **string**; Value: **string**): Boolean;

**bool \_\_fastcall** ONVIF\_SetStr(System::wchar\_t \*ParamIdentifier, System::wchar\_t \*Value);

#### Description

Used to pass settings to the Datstead [RTSP/RTMP/HTTP/ONVIF DirectShow Source Filter](#)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastError](#) [GetLastError](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Eliminate the Struggles of Documentation with a Help Authoring Tool](#)

## ONVIFCancelDiscovery

### TVideoGrabber.ONVIFCancelDiscovery

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Cancels the ONVIF discovery

#### Declaration

**function** ONVIFCancelDiscovery: Boolean;

**bool** \_\_fastcall ONVIFCancelDiscovery();

#### Description

Invoke it to ancel a [ONVIFDiscoverCameras](#) [Multicast](#) or [ONVIFDiscoverCameras](#) [IPRange](#) background process currently running

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastError](#) [GetLastError](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## ONVIFDeviceInfo

### TVideoGrabber.ONVIFDeviceInfo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieve ONVIF information

#### Declaration

**function** ONVIFDeviceInfo (ONVIFDeviceInfoType: TONVIFDeviceInfo): **string**;

**const** wchar\_t \* \*\_ONVIFDeviceInfo (void \*TVGObject, TONVIFDeviceInfo ONVIFDeviceInfoType);

#### Description

Returns the specified ONVIF information as a string.

Set [ONVIFDeviceInfoType](#) with the [type of ONVIF information](#) desired.

## See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras\\_IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsyncStatus](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

## ONVIFDiscoverCameras\_IPRange

### TVideoGrabber.ONVIFDiscoverCameras\_IPRange

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

ONVIF discovery from a range of IP addresses

#### Declaration

**function** ONVIFDiscoverCameras\_IPRange (First\_IP: **string**; Last\_IP: **string**; timeout\_seconds\_or\_0\_for\_default: LongInt): Boolean;

**bool** \_\_fastcall ONVIFDiscoverCameras\_IPRange(System::UnicodeString First\_IP, System::UnicodeString Last\_IP, System::LongInt timeout\_seconds\_or\_0\_for\_default);

#### Description

Scans the network for ONVIF cameras between the ip addresses specified, e.g. from 192.168.0.1 to 192.168.0.254

See [ONVIF Cameras Discovery](#)

## See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras\\_IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## ONVIFDiscoverCameras\_Multicast

### TVideoGrabber.ONVIFDiscoverCameras\_Multicast

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

ONVIF discovery from a Multicast request

#### Declaration

**function** ONVIFDiscoverCameras\_Multicast (timeout\_seconds\_or\_0\_for\_default: LongInt): Boolean;

**bool** \_\_fastcall ONVIFDiscoverCameras\_Multicast(System::LongInt timeout\_seconds\_or\_0\_for\_default);

#### Description

Scans the network for ONVIF cameras by broadcasting a multicast request. Note that the network switch(es) must support IGMP snooping.

See [ONVIF Cameras Discovery](#)

## See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF](#) [GetStr](#) [ONVIF](#) [SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## ONVIFEnumCamerasDiscovered

### TVideoGrabber.ONVIFEnumCamerasDiscovered

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the ONVIF cameras discovered

#### Declaration

**function** ONVIFEnumCamerasDiscovered (CameraIndex: LongInt; **var** CameraType: **string**; **var** CameraONVIFUrl: **string**): Boolean;

**bool** \_\_fastcall ONVIFDiscoverCameras\_IPRange(System::UnicodeString First\_IP, System::UnicodeString Last\_IP, System::LongInt timeout\_seconds\_or\_0\_for\_default);

#### Description

Retrieve the camera type and ONVIF service URL by specifying the index of the IP cameras in the 0..n-1 range.

The camera count is returned by each call of the [OnONVIFDiscoveryCompletedNotification](#) event.

Note that it is not required to know the camera count, just invoke ONVIFEnumCamerasDiscovered starting from CameraIndex = 0, by increasing it at each call.

The function call returns false when the CameraIndex = the number of cameras discovered.

See [ONVIF Cameras Discovery](#)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF](#) [GetStr](#) [ONVIF](#) [SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Documentation Workflow with HelpNDoc's Intuitive UI](#)

## ONVIFPTZGetLimits

### TVideoGrabber.ONVIFPTZGetLimits

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the limit PTZ positions

#### Declaration

**function** ONVIFPTZGetLimits (**out** Pan\_Min: Double; **out** Pan\_Max: Double; **out** Tilt\_Min: Double; **out** Tilt\_Max: Double; **out** Zoom\_Min: Double; **out** Zoom\_Max: Double): Boolean;

BOOL \_ONVIFPTZGetLimits (**void** \*TVGObject, **double**\*Pan\_Min, **double**\*Pan\_Max, **double**\*Tilt\_Min,

**double\***Tilt\_Max, **double\***Zoom\_Min, **double\***Zoom\_Max);

### Description

Returns the limit PTZ positions (min and max) as double values

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastError](#) [GetLastError](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsyncStatus](#)

Created with the Standard Edition of HelpNDoc: [Easily create Epub books](#)

## ONVIFPTZGetPosition

### TVideoGrabber.ONVIFPTZGetPosition

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the current PTZ position

### Declaration

**function** ONVIFPTZGetPosition (**out** Pan: Double; **out** Tilt: Double; **out** Zoom: Double; **out** UTCTime: LargeInteger; **out** IsMoving: LongInt): Boolean;

BOOL \_ONVIFPTZGetPosition (**void** \*TVGObject, **double\***Pan, **double\***Tilt, **double\***Zoom, \_\_int64\*UTCTime, **int\***IsMoving);

### Description

Returns the current PTZ position as double values

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastError](#) [GetLastError](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsyncStatus](#)

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

## ONVIFPTZPreset

### TVideoGrabber.ONVIFPTZPreset

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Manage the presets

### Declaration

**function** ONVIFPTZPreset (PresetAction: **string**; PresetName: **string**): Boolean;

BOOL \_ONVIFPTZPreset (**void** \*TVGObject, **const wchar\_t \***PresetAction, **const wchar\_t \***PresetName);

### Description

Creates, remove or goto the specified preset, if the camera supports the presets

PresetAction can be "CREATE", "REMOVE" or "DELETE"

PresetName can be any choosen name, however some cameras support only their own predefined names



of presets

To create a preset, move the camera to the desired position with `ONVIFPTZStartContinuousMove` or `ONVIFPTZSetPosition`, then invoke `ONVIFPTZPreset` ("CREATE", ThePresetName) to save the current position to the specified preset.

Then the camera can be positioned to this position when needed by invoking `ONVIFPTZPreset` ("GOTO", ThePresetName)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsyncStatus](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## ONVIFPTZSendAuxiliaryCommand

### TVideoGrabber.ONVIFPTZSendAuxiliaryCommand

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Send an auxiliary command

#### Declaration

**function** ONVIFPTZSendAuxiliaryCommand (AuxiliaryCommand: **string**): Boolean;

BOOL \_ONVIFPTZSendAuxiliaryCommand (**void** \*TVGObject, **const wchar\_t** \* \*AuxiliaryCommand);

#### Description

Used to send a manufacturer-specific command to the camera.

Read the camera documentation to see what auxiliary commands the camera supports.

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsyncStatus](#)

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## ONVIFPTZSetPosition

### TVideoGrabber.ONVIFPTZSetPosition

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Set a new PTZ position

#### Declaration

**function** ONVIFPTZSetPosition (Pan: Double; Tilt: Double; Zoom: Double; SpeedRatio: Double; IsRelative: Boolean): Boolean;

BOOL \_ONVIFPTZSetPosition (**void** \*TVGObject, **double** Pan, **double** Tilt, **double** Zoom, **double** SpeedRatio, BOOL IsRelative);

#### Description



[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#)  
[TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastErrorMessages](#) [OnONVIFDiscoveryCompletedNotification](#)  
[ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#)  
[ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#)  
[ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZStartMove](#)  
[ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsyncStatus](#)

## ONVIFPTZStartMove

## TVideoGrabber Methods

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#)  
[TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastErrorMessages](#) [OnONVIFDiscoveryCompletedNotification](#)  
[ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#)  
[ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#)  
[ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#)  
[ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsyncStatus](#)

## ONVIFPTZStopMove

## TVideoGrabber Methods

522 / 761

**function** ONVIFPTZStopMove (PTZType: **string**): Boolean;

BOOL \_ONVIFPTZStopMove (**void** \*TVGObject, **const wchar\_t** \* PTZType);

#### Description

Stops moving the specified PTZ

PTZType can be "Pan", "Tilt" or "Zoom"

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastError](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFSnapShot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCCommand](#) [SetIPCSetting](#)

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## ONVIFSnapShot

### TVideoGrabber.ONVIFSnapShot

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Capture a JPEG snapshot from an ONVIF IP camera

#### Declaration

**function** ONVIFSnapShot(OnRawVideoSampleCallbackEnabled: Boolean; SaveToFile: Boolean; FileName: **string**): Boolean;

BOOL ONVIFSnapShot(**bool** OnRawVideoSampleCallbackEnabled, **bool** SaveToFile, System::wchar\_t \*FileName);

#### Description

Used to capture a JPEG snapshot from an IP camera supporting the ONVIF protocol

bool OnRawVideoSampleCallbackEnabled: if true, the OnRawVideoSample event occurs, it is possible to get a direct access to the the buffer and buffer size parameters of the event, the buffer contains the JPEG image in memory.

bool SaveToFile: if true, the JPEG image is saved to the FileName specified

FileName: path/file name of the JPEG image to save

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [ExtraDLLPath](#) [GetLastError](#) [GetLastErrorMessage](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [OpenURLAsyncStatus](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EPub generator](#)

## OpenDVD

### TVideoGrabber.OpenDVD

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Opens a DVD in playback mode

## Declaration

**function** OpenDVD: LongBool;

BOOL \_\_fastcall OpenDVD(void)

Function OpenDVD as Boolean

## Description

Used to opens a DVD in playback mode

If [DVDTitle](#) = 0, the DVD menu appears when opening the DVD playback.

To start directly the playback of a DVD title, set DVDTitle with the number of the title.

E.g.:

```
VideoGrabber.PlayerFileName = "D:
VideoGrabber.DVDTitle = 0
VideoGrabber.OpenDVD ()
```

will open the DVD menu.

```
VideoGrabber.PlayerFileName = "E:
VideoGrabber.DVDTitle = 1
VideoGrabber.OpenDVD ()
```

will start playing directly the 1st title.

## See Also

[DVDInfo](#) [DVDTitle](#) [PlayerFileName](#)

---

Created with the Standard Edition of HelpNDoc: [Experience the Power and Ease of Use of a Help Authoring Tool](#)

---

## OpenPlayer

### TVideoGrabber.OpenPlayer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Opens a video clip.

## Declaration

**function** OpenPlayer: Boolean;

bool \_\_fastcall OpenPlayer(void)

Function OpenPlayer() As Boolean

## Description

Opens the video clip specified by [PlayerFileName](#) .

## See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#)

[PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#)  
[PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#)  
[PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#)  
[RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#)  
[SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## OpenPlayerAtFramePositions

### TVideoGrabber.OpenPlayerAtFramePositions

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Opens a video clip within a specified start frame and stop frame.

#### Declaration

**function** OpenPlayerAtFramePositions (StartFrame: LargeInteger; StopFrame: LargeInteger; KeepBounds: Boolean; CloseAndReopenIfAlreadyOpened: Boolean): Boolean;

**bool** \_\_fastcall OpenPlayerAtFramePositions(\_\_int64 StartFrame, \_\_int64 StopFrame, **bool** KeepBounds, **bool** CloseAndReopenIfAlreadyOpened)

Function OpenPlayerAtFramePositions(StartFrame As Double, StopFrame As Double, KeepBounds as Boolean, CloseAndReopenIfAlreadyOpened as Boolean)As Boolean

#### Description

Opens a video clip that will start and stop playing within boundaries between specified start and stop frames.

#### StartFrame:

opens the clip starting from this frame number (0 = from beginning)

#### StopFrame:

stops playing the clip at this frame number (0 = until the end)

#### KeepBounds:

##### - true:

- . when invoking StopPlayer the position returns to the StartFrame location
- . when the clip is paused, it is not possible to set a position before StartFrame or after StopFrame

##### - false:

- . when invoking StopPlayer the position returns at the frame number 1
- . when the clip is paused, it is possible to set a position before StartFrame or after StopFrame

#### CloseAndReopenIfAlreadyOpened:

##### - true:

if OpenPlayerAtFramePositions is invoked again the clip is reopened

##### - false:

if OpenPlayerAtFramePositions is invoked again the start and stop positions are updated to the specified positions

#### Notes:

- when the video clip has already been opened and the **CloseAndReopenIfAlreadyOpened** parameter is false, invoking this function lets you modify the playing boundaries, without closing/reopening the video clip
- to specify only a start frame and let the clip play until the end, set StopFrame = 0
- to specify only a stop frame and let the clip play from the beginning, set StartFrame = 0
- to reset the normal clip boundaries, set StopFrame = 0 and StartFrame = 0

E.g. OpenPlayerAtFramePositions (100, 900) plays the clip within the 100th and the 900th frames.

**See Also**

[Player features](#)
[TOnPlayerStateChanged](#)
[TPlayerState](#)
[TOnPlayerBufferingData](#)
[AudioChannelRenderMode](#)
[AudioStreamNumber](#)
[AutoStartPlayer](#)
[AVIDuration](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[AVIInfo2](#)
[ClosePlayer](#)
[FastForwardPlayer](#)
[IsPlayerAudioStreamAvailable](#)
[IsPlayerVideoStreamAvailable](#)
[Last\\_Clip\\_Played](#)
[MP4NeedsReindexing](#)
[OnPlayerBufferingData](#)
[OnPlayerEndOfStream](#)
[OnPlayerStateChanged](#)
[OnPlayerUpdateTrackbarPosition](#)
[OpenPlayer](#)
[OpenPlayerAtTimePositions](#)
[PausePlayer](#)
[PlayerAudioCodec](#)
[PlayerAudioRendering](#)
[PlayerDuration](#)
[PlayerDVSize](#)
[PlayerFastSeekSpeedRatio](#)
[PlayerFileName](#)
[PlayerForcedCodec](#)
[PlayerFrameCount](#)
[PlayerFramePosition](#)
[PlayerFrameRate](#)
[PlayerFrameStep](#)
[PlayerRefreshPausedDisplay](#)
[PlayerRefreshPausedDisplayFrameRate](#)
[PlayerSpeedRatio](#)
[PlayerTimePosition](#)
[PlayerTrackBar](#)
[PlayerTrackBarScale](#)
[PlayerTrackBarSynchronise](#)
[PlayerVideoCodec](#)
[RewindPlayer](#)
[RunPlayer](#)
[RunPlayerBackwards](#)
[ShowDialog](#)
[SourceStream](#)
[StopPlayer](#)
[SynchronizationRole](#)
[Synchronized](#)
[VideoHeight](#)
[PreferredAspectRatio](#)
[VideoPlayableWhileRecording](#)
[VideoWidth](#)
[PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Spot and Fix Problems in Your Documentation with HelpNDoc's Project Analyzer

**OpenPlayerAtTimePositions****TVideoGrabber.OpenPlayerAtTimePositions**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Opens a video clip within a specified start time and stop time.

**Declaration**

**function** OpenPlayerAtTimePositions (StartTime: LargeInteger; StopTime: LargeInteger; KeepBounds: Boolean; CloseAndReopenIfAlreadyOpened: Boolean): Boolean;

**bool** \_\_fastcall OpenPlayerAtTimePositions(\_\_int64 StartTime, \_\_int64 StopTime, **bool** KeepBounds, **bool** CloseAndReopenIfAlreadyOpened)

Function OpenPlayerAtTimePositions(StartTime As Double, StopTime As Double, CloseAndReopenIfAlreadyOpened as String) As Boolean

**Description**

Opens a video clip that will start and stop playing within boundaries between specified start and stop time. StartTime and StopTime are expressed in 100ns units (e.g. 2.5 seconds = 25000000).

**StartTime:**

opens the clip starting from this frame time (0 = from beginning)

**StopTime:**

stops playing the clip at this frame time (0 = until the end)

**KeepBounds:****- true:**

- . when invoking StopPlayer the position returns to the StartTime position
- . when the clip is paused, it is not possible to set a position before StartTime or after StopTime

**- false:**

- . when invoking StopPlayer the position returns at the frame time 0
- . when the clip is paused, it is possible to set a position before StartTime or after StopTime

**CloseAndReopenIfAlreadyOpened:****- true:**

if OpenPlayerAtTimePositions is invoked again the clip is reopened

**- false:**

if OpenPlayerAtTimePositions is invoked again the start and stop positions are updated to the specified positions

#### Notes:

- when the video clip has already been opened and the **CloseAndReopenIfAlreadyOpened** parameter is false, invoking this function lets you modify the playing boundaries, without closing/reopening the video clip
- to specify only a start time and let the clip play until the end, set `StopTime = 0`
- to specify only a stop time and let the clip play from the beginning, set `StartTime = 0`
- to reset the normal clip boundaries, set `StopTime = 0` and `StartTime = 0`

E.g. `OpenPlayerAtFramePositions (30000000, 60000000)` plays the clip within 3 seconds and the 6 seconds (starting from the beginning).

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

## OpenURLAsyncStatus

### TVideoGrabber.OpenURLAsyncStatus

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

returns the status of the URL currently being opened asynchronously

#### Declaration

**function** OpenURLAsyncStatus: TOpenURLAsyncStatus;

TOpenURLAsyncStatus \_\_**fastcall** OpenURLAsyncStatus();

#### Description

meaning of the values here: [TOpenURLAsyncStatus](#)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsync](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## PausePlayer

### TVideoGrabber.PausePlayer

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

Pauses a video clip currently playing back.

### Declaration

**procedure** PausePlayer;

**void** \_\_fastcall PausePlayer(**void**)

Sub PausePlayer()

### Description

Used to pause a [PlayerFileName](#) video clip currently playing back.  
Call [RunPlayer](#) to resume a paused clip.

### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Produce online help for Qt applications](#)

## PausePreview

### TVideoGrabber.PausePreview

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

Pauses running preview.

### Declaration

**function** PausePreview: Boolean;

**bool** \_\_fastcall PausePreview(**void**)

Function PausePreview() As Boolean

### Description

Used to pause preview, previously started with [StartPreview](#).

Can be invoked also during recording (started with [StartRecording](#)) to pause the recording AND the preview. Then invoke [ResumePreview](#) to go on recording AND previewing (see also [Pause/resume during recording](#))

### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)



## PauseRecording

### TVideoGrabber.PauseRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Pauses the current recording.

#### Declaration

**function** PauseRecording: Boolean;

**bool** \_\_fastcall PauseRecording(**void**)

Function PauseRecording() As Boolean

#### Description

Used to pause the current recording, previously started by [StartRecording](#) .

[RecordingCanPause](#) must be enabled to use this feature.

Returns true if the recording has been successfully paused.

recording can be resumed with [ResumeRecording](#) .

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

## PlayerFrameStep

### TVideoGrabber.PlayerFrameStep

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Steps the video clip n frames forward

#### Declaration

**function** PlayerFrameStep (FrameCount: LongInt): Boolean;

**bool** \_\_fastcall PlayerFrameStep(**int** FrameCount);

Function PlayerFrameStep as Bool

#### Description

Used to step n frames forward when the video clip is paused.

The FrameCount parameter specifies the number of frames to skip.

- if FrameCount is 1, the graph steps forward one frame.
- if dwFrames is a number n greater than 1, the graph skips n - 1 frames and shows the nth frame.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Encrypted, Password-Protected PDFs](#)

## Playlist

### TVideoGrabber.Playlist

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Used for building and playing playlists.

#### Declaration

**function** Playlist (PlaylistAction: TPlaylist; VideoClip: **String**): Boolean;

**int** \_\_fastcall PointGreyConfig(TPointGreyConfig ConfigType, **unsigned** &PGRActionValue, **unsigned** ulRegister, **unsigned** ulMode, **unsigned** ulLeft, **unsigned** ulTop, **unsigned** ulWidth, **unsigned** ulHeight, **unsigned** ulPercentage, **unsigned** ulFormat);

**function** Playlist (PlaylistAction as TxPlaylist, VideoClip as string)

#### Description

This function is used for building and playing playlists in the player.

- the PlaylistAction parameter specifies the action
- the VideoClip parameter specifies the video clip on which the action is applied.

See the "[Using the playlist](#)" chapter in the **Player** section of the user guide.

#### See Also

[TPlaylist](#) [Video formats](#) [GetPlaylist](#) [IsPlaylistActive](#) [OnPlayerEndOfPlaylist](#) [PlaylistIndex](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

## PointGreyConfig

### TVideoGrabber.PointGreyConfig

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to configure a PointGrey camera

**Declaration**

**function** PointGreyConfig (ConfigType: TPointGreyConfig; **var** PGRActionValue: LongWord; ulRegister: LongWord; ulMode: LongWord; ulLeft: LongWord; ulTop: LongWord; ulWidth: LongWord; ulHeight: LongWord; ulPercentage: LongWord; ulFormat: LongWord): LongInt;

**int \_\_fastcall** PointGreyConfig(TPointGreyConfig ConfigType, **unsigned** &PGRActionValue, **unsigned** ulRegister, **unsigned** ulMode, **unsigned** ulLeft, **unsigned** ulTop, **unsigned** ulWidth, **unsigned** ulHeight, **unsigned** ulPercentage, **unsigned** ulFormat);

function PointGreyConfig (ConfigType as TPointGreyConfig, ByRef PGRActionValue as Long, ulRegister as Long, ulMode as Long, ulLeft as Long, ulTop as Long, ulWidth as Long, ulHeight as Long, ulPercentage as Long, ulFormat as LongWord) as Long

**Description**

Used to configure PointGrey cameras using the PointGrey's "FlyStream" driver

The 1st PointGreyConfig parameter (type [TPointGreyConfig](#)) specify the type of action applied. The use of the other parameters depend on the PointGreyConfig parameter:

**pgr\_SetRegister**

sets the register specified in **ulRegister** with the value specified in **PGRActionValue**  
*the other parameters are ignored*

**pgr\_GetRegister**

get the register specified in **ulRegister**, the value is returned by **PGRActionValue**  
*the other parameters are ignored*

**pgr\_SetBufferSize**

sets the buffer size with the value specified in **PGRActionValue**  
*the other parameters are ignored*

**pgr\_GetBufferSize**

gets the buffer size, the value is returned by **PGRActionValue**  
*the other parameters are ignored*

**pgr\_SetFormat**

sets the format with the values specified in **ulMode**, **ulLeft**, **ulTop**, **ulWidth**, **ulHeight**, **ulPercentage**, **ulFormat**  
*the other parameters are ignored*

---

Created with the Standard Edition of HelpNDoc: [Full-featured Documentation generator](#)

---

**PutMiscDeviceControl****TVideoGrabber.PutMiscDeviceControl**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to set specific values or states on some video capture devices.

**Declaration**

**function** PutMiscDeviceControl (MiscDeviceControl: TMiscDeviceControl; **Index**: LongInt; **Value**: LongInt): Boolean;

**bool \_\_fastcall** PutMiscDeviceControl(TMiscDeviceControl MiscDeviceControl, **int** Index, **int** Value);

Function PutMiscDeviceControl (MiscDeviceControl as TMiscDeviceControl, Index as Long, Value as Long) as Boolean

**Description**

Used to set specific values or states on some video capture devices that support them, e.g. the GPIO, VPD, etc...

The use of this function is explained in the [Miscellaneous device control](#) chapter.

## See Also

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## RecordingKBytesWrittenToDisk

### TVideoGrabber.RecordingKBytesWrittenToDisk

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the file size of the current recording.

#### Declaration

**function** RecordingKBytesWrittenToDisk: LongWord;

**unsigned \_\_fastcall** RecordingKBytesWrittenToDisk(**void**);

function RecordingKBytesWrittenToDisk as Long

#### Description

While recording, this function returns the file size in KB of the file currently beeing recorded.

This function reports the current (growing) file size in KB while the recording is running.

Invoke this function periodically (e.g. every 10 seconds) to retrieve on the fly the current file size of the video clip being recorded.

When the recording ends, invoke [RecordingKBytesWrittenToDisk](#) from the [OnRecordingCompleted](#) event to get the final file size of the recorded clip.

**(note: due to the recording file buffer, the real size at a given time is always a bit larger than the size reported by this function)**

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## RecordToNewFileNow

### TVideoGrabber.RecordToNewFileNow

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Creates a new file during recording.

#### Declaration

**function** RecordToNewFileNow (NewRecordingFileName: **string**; ResetStreamTime: Boolean): Boolean;

**bool \_\_fastcall** RecordToNewFileNow (wchar\_t \*NewRecordingFileName, **bool** ResetStreamTime);

Function RecordToNewFileNow((NewRecordingFileName as String, ResetStreamTime as Boolean) As Boolean

#### Description

During recording, invoking **RecordToNewFileNow** closes the current recording file, and creates a new recording file according to the [RecordingFileName](#) property.

- if [RecordingFileName](#) is empty, a new file name is generated automatically according to the [StoragePath](#) , [AutoFileName](#) and [AutoFilePrefix](#) .

- if [RecordingFileName](#) specifies a file name, this file name will be used to create the new file.

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

## RefreshDevicesAndCompressorsLists

### TVideoGrabber.RefreshDevicesAndCompressorsLists

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Refreshes the devices and compressors list.

#### Declaration

**procedure** RefreshDevicesAndCompressorsLists;

**void \_\_fastcall** RefreshDevicesAndCompressorsLists (**void**)

Sub RefreshDevicesAndCompressorsLists

#### Description

Used mainly to refresh the [VideoCompressors](#) and [AudioCompressors](#) list, e.g. after installing/uninstalling video compressors or audio compressors.

Calling this procedure refreshes also the [AudioDevices](#) and [VideoDevices](#) lists. However usually it is not useful to invoke this procedure for video capture devices or audio capture devices, because when connecting/disconnecting them, the corresponding lists are refreshed automatically and the [OnDeviceArrivalOrRemoval](#) event occurs to notify the list have been updated.

#### See Also

[Recording methods and properties](#) [WDM drivers](#) [TCompressionType](#) [TOnDeviceArrivalOrRemoval](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [AutoConnectRelatedPins](#) [CompressionMode](#) [CompressionType](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [GetVideoCompressionSettings](#) [IsDVDDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnDeviceArrivalOrRemoval](#) [OnDeviceLost](#) [OnNoVideoDevices](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [OnVideoDeviceSelected](#) [ResetVideoDeviceSettings](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [ShowDialog](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

## RefreshPlayerOverlays

### TVideoGrabber.RefreshPlayerOverlays

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

must be invoked to refresh the graphic or text overlays when the video clip is paused

#### Declaration

**procedure** RefreshPlayerOverlays;

**void** \_\_fastcall RefreshPlayerOverlays(**void**)

Sub RefreshPlayerOverlays()

#### Description

Invoking this method when the video clip is paused raises the OnFrameOverlayUsing... events and redraw the current video frame.

Invoke it if you are e.g. performing free hand drawing with the mouse over the video frames while the clip is paused, and you want the drawing to be applied.

Look at the "free hand drawing" sample code in the OnFrameOverlayUsingDC event of the MainDemo project.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

## ResetVideoDeviceSettings

### TVideoGrabber.ResetVideoDeviceSettings

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)



**ber**

Resets the current settings of the video capture devices

**Declaration**

**function** ResetVideoDeviceSettings: Boolean;

**bool** \_\_fastcall ResetVideoDeviceSettings(**void**)

Function ResetVideoDeviceSettings() As Boolean

**Description**

Used to reset the settings of the current video capture device.

Just invoke this function one time while the component is inactive, then the default settings will be used the next time the preview or recording is started.

**See Also**

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

**ResumePreview****TVideoGrabber.ResumePreview**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Resumes paused preview.

**Declaration**

**function** ResumePreview: Boolean;

**bool** \_\_fastcall ResumePreview(**void**)

Function ResumePreview() As Boolean

**Description**

Used to resume preview, previously paused with [PausePreview](#).

Can be invoked also during a recording (started with [StartRecording](#)) to resume a recording **AND** a preview if they have been paused by invoking [PausePreview](#). (see also [Pause/resume during recording](#))

**See Also**

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

**ResumeRecording****TVideoGrabber.ResumeRecording**
[Prev](#)
[Next](#)



## TVideoGrabber [Methods](#)

Resumes an recording, currently in a paused state.

### Declaration

**function** ResumeRecording: Boolean;

**bool** \_\_fastcall ResumeRecording(**void**)

Function ResumeRecording() As Boolean

### Description

Used to resume an recording, currently paused by a [PauseRecording](#) call. [RecordingCanPause](#) must be enabled to use this feature.

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

## RetrieveInitialXYAfterRotation

### TVideoGrabber.RetrieveInitialXYAfterRotation

[Prev](#)
[Next](#)

#### [TVideoGrabber](#) [Methods](#)

Retrieves the original frame coordinates after a rotation.

### Declaration

**procedure** RetrieveInitialXYAfterRotation (X, Y: LongInt; var OriginalX: LongInt; var OriginalY: LongInt);

**void** \_\_fastcall RetrieveInitialXYAfterRotation (**int** X, **int** Y, **int** &OriginalX, **int** &OriginalY)

Sub RetrieveInitialXYAfterRotation (X as Long, Y as Long, OriginalX as Variant, OriginalY as Variant)

### Description

Used to retrieve the initial frame coordinates after a [VideoProcessing\\_Rotation](#) as been applied and [OverlayAfterTransform](#) is enabled.

This procedure is used mainly to make a drawing from the [OnFrameOverlayUsingDC](#) event that is not rotated when [OverlayAfterTransform](#) is enabled.

E.g.:

```
procedure TfrmMainForm.VideoGrabberFrameOverlayUsingDC(Sender: TObject; Dc: HDC; FrameNum
```

```

var
  Canvas: TCanvas;
  SrcX, SrcY: LongInt;
  DestX, DestY: LongInt;
begin
  Canvas := TCanvas.Create;
  Canvas.Handle := Dc;
  Canvas.Pen.Color := clWhite;
  VideoGrabber.RetrieveInitialXYAfterRotation (100, 100, SrcX, SrcY);
  VideoGrabber.RetrieveInitialXYAfterRotation (100, 200, DestX, DestY);
  Canvas.MoveTo (SrcX, SrcY);
  Canvas.Pixels[SrcX, SrcY] := clWhite;
  Canvas.LineTo (DestX, DestY);
  Canvas.Free;
end;

```

**See Also**

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [VideoProcessing\\_Brightness](#)  
[VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#)  
[VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#)  
[VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#)  
[VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Make your documentation accessible on any device with HelpNDoc](#)

**RewindPlayer****TVideoGrabber.RewindPlayer**[Prev](#)[Next](#)

[TVideoGrabber](#)   [Methods](#)  
[ber](#)

Plays a clip backwards.

**Declaration**

**procedure** RewindPlayer;

**void** \_\_fastcall RewindPlayer(**void**)

Sub RewindPlayer()

**Description**

Used to play a video clip backwards.

The backwards speed is the normal speed x [PlayerFastSeekRatio](#) .

Restrictions:

- this feature is available only with seekable (indexed) AVI clips.
- according to the clip index and the disk access, playing backwards may be jerky.

**See Also**

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#)  
[AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#)  
[FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#)  
[MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#)  
[OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#)  
[PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#)  
[PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#)  
[PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#)  
[PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#)

[PlayerVideoCodec](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#)  
[SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Full-featured Help generator

## RunPlayer

### TVideoGrabber.RunPlayer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts playing a video clip.

#### Declaration

**procedure** RunPlayer;

**void** \_\_fastcall RunPlayer(**void**)

Sub RunPlayer()

#### Description

Used to play a [PlayerFileName](#) video clip opened by [OpenPlayer](#) .

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#)  
[AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#)  
[FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#)  
[MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#)  
[OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#)  
[PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#)  
[PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#)  
[PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#)  
[PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#)  
[PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#)  
[SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: Create cross-platform Qt Help files

## RunPlayerBackwards

### TVideoGrabber.RunPlayerBackwards

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts playing a video clip backwards.

#### Declaration

**procedure** RunPlayerBackwards;

**void** \_\_fastcall RunPlayerBackwards(**void**)

Sub RunPlayerBackwards()

#### Description

Used to play backwards a [PlayerFileName](#) video clip opened by [OpenPlayer](#) .

#### Restrictions:

- this feature is available only with seekable (indexed) AVI clips.
- according to the clip index and the disk access, playing backwards may be jerky.

**See Also**

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's Easy-to-Use Tool](#)

**SaveCompressorSettingsToDataString****TVideoGrabber.SaveCompressorSettingsToDataString**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Saves the compressor settings to a data string

**Declaration**

**function** SaveCompressorSettingsToDataString (IsVideoCompressor: boolean; CompressorIndex: LongInt): **string**;

wchar\_t \* **fastcall** SaveCompressorSettingsToDataString (**bool** IsVideoCompressor, **int** CompressorIndex: LongInt);

function SaveCompressorSettingsToDataString (IsVideoCompressor as Boolean, CompressorIndex as Long) As String

**Description**

Used to save the current compressor settings to a data string.

The compressor settings depend on the current [VideoCompressor](#) and [RecordingMethod](#), so be sure to first select the RecordingMethod() and VideoCompressor before saving its settings.

The **IsVideoCompressor** parameter must be set to **true** for a video compressor, or **false** for an audio compressor.

The Compressor parameter index must be the [VideoCompressor](#) or [AudioCompressor](#) index of the related compressor in the [VideoCompressors](#) or [AudioCompressors](#) lists.

See [Saving and restoring compressor settings programmatically](#) for sample code.

**See Also**

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TCompressionType](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TOnVideoCompressionSettings](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [CompressionMode](#) [CompressionType](#) [Encoder](#) [SetInt](#) [GetVideoCompressionSettings](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [PauseRecording](#)

[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[RefreshDevicesAndCompressorsLists](#)
[ResumeRecording](#)
[SetMultiplexerFilterByName](#)
[SetVideoCompressionDefaults](#)
[SetVideoCompressionSettings](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoCompression](#)
[KeyFrameRate](#)
[VideoCompression](#)
[PFramesPerKeyFrame](#)
[VideoCompression](#)
[Quality](#)
[VideoCompression](#)
[WindowSize](#)
[VideoCompressor](#)
[VideoCompressorIndex](#)
[VideoCompressorName](#)
[VideoCompressors](#)
[VideoCompressorsCount](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

## SaveCompressorSettingsToTextFile

### TVideoGrabber.SaveCompressorSettingsToTextFile

[Prev](#)

[Next](#)

[TVideoGrabber](#)
[Methods](#)

Saves the current compressor settings to a text file.

#### Declaration

**function** SaveCompressorSettingsToTextFile (IsVideoCompressor: boolean; CompressorIndex: LongInt; FileName: **string**): Boolean;

**bool** \_\_fastcall SaveCompressorSettingsToTextFile (**bool** IsVideoCompressor, **int** CompressorIndex: LongInt, wchar\_t \*FileName);

function SaveCompressorSettingsToTextFile (IsVideoCompressor as Boolean, CompressorIndex as Long, FileName as String) As Boolean

#### Description

Used to save the current compressor settings to a text file.

The compressor settings depend on the current [VideoCompressor](#) and [RecordingMethod](#), so be sure to first select the RecordingMethod() and VideoCompressor before saving its settings.

The **IsVideoCompressor** parameter must be set to **true** for a video compressor, or **false** for an audio compressor.

The Compressor parameter index must be the [VideoCompressor](#) or [AudioCompressor](#) index of the related compressor in the [VideoCompressors](#) or [AudioCompressors](#) lists.

See [Saving and restoring compressor settings programmatically](#) for sample code.

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## ScreenRecordingUsingCoordinates

### TVideoGrabber.ScreenRecordingUsingCoordinates

[Prev](#)

[Next](#)

[TVideoGrabber](#)
[Methods](#)

Limits the screen recording to an area

#### Declaration

**function** ScreenRecordingUsingCoordinates (Enabled: Boolean; scLeft, scTop, scWidth, scHeight: LongInt): Boolean;

**bool \_\_fastcall** ScreenRecordingUsingCoordinates(**bool** Enabled, **int** scLeft, **int** scTop, **int** scWidth, **int** scHeight);

Function ScreenRecordingUsingCoordinates (Enabled as Boolean, scTop, scWidth, scHeight as Long) as Boolean

### Description

Used to limit the screen recording to a part of the screen.

**Enabled** : activates the limitation (if disabled, the whole screen is recorded)

**scLeft** : left coordinate of the screen area to record

**scTop** : top coordinate of the screen area to record

**scWidth** : width of the screen area to record

**scHeight** : heigth of the screen area to record

Remarks:

- the scLeft and scTop location may be modified dynamically while previewing or recording
- this feature is inactive if the recording of a window has been activated with [SetWindowRecordingByName](#) or [SetWindowRecordingByHandle](#)

### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## SendCameraCommand

### TVideoGrabber.SendCameraCommand

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sends Pan and Tilt commands to the current camera

### Declaration

**function** SendCameraCommand (Pan: LongInt; Tilt: LongInt; Relative: Boolean): Boolean;

**bool \_\_fastcall** SendCameraCommand (**int** Pan, **int** Tilt, **bool** Relative);

function SendCameraCommand Pan as Long, Tilt as Long, Relative as Boolean)

### Description

Used to send Pan and Tilt commands to the camera currently selected as [VideoDevice](#).

Pass as Pan and Tilt the number of steps to perform. Use:

- . negative values to pan left
- . positive values to pan right
- . negative values to tilt down
- . positive values to tilt up
- . if Relative is true, the steps are applied to the current position of the camera
- . if Relative is false, the steps are applied from the origin position of the camera.

E.g.:

- **one step left, from the current position:**

VideoGrabber.SendCameraCommand (-1, 0, true);

- **two steps right, from the current position:**

```
VideoGrabber.SendCameraCommand (2, 0, true);
```

**- one step up, from the current position:**

```
VideoGrabber.SendCameraCommand (0, 1, true);  
end;
```

**- one step down, from the origin:**

```
VideoGrabber.SendCameraCommand (0, -1, false);
```

**- 2 steps right and 1 step up from the current position**

```
VideoGrabber.SendCameraCommand (2, 1, true);
```

**- back to the origin:**

```
VideoGrabber.SendCameraCommand (0, 0, false);
```

**See Also**

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SetCameraControl](#) [SetCameraExposure](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## SendDVCommand

### TVideoGrabber.SendDVCommand

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sends a command to the VCR.

#### Declaration

```
function SendDVCommand(DVCommand: TDVCommand): Boolean;
```

```
bool __fastcall SendDVCommand(TDVCommand DVCommand)
```

Function SendDVCommand(DVCommand As TxDVCommand) As Boolean

#### Description

Sends the specified [TDVCommand](#) transport command to the DV device.  
Returns true upon success.

**See Also**

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

## SendImageToVideoFromBitmaps

### TVideoGrabber.SendImageToVideoFromBitmaps

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to pass the image or bitmap handle to the VideoFromJPEGsOrBitmaps function

#### Declaration

```
function SendImageToVideoFromBitmaps (ImageFilePath: string; BitmapHandle: LongInt;  
CanFreeBitmapHandle: Boolean; EndOfData: Boolean): boolean;
```



```
bool __fastcall SendImageToVideoFromBitmaps(wchar_t *ImagePath, int BitmapHandle, bool CanFreeBitmapHandle, bool EndOfData);
```

function SendImageToVideoFromBitmaps (ImagePath as string, BitmapHandle as Long, CanFreeBitmapHandle as Boolean, EndOfData as Boolean) as Boolean

### Description

When [VideoSource](#) = vs\_JPEGsOrBitmaps, TVideoGrabber needs the video frames to be passed by SendImageToVideoFromBitmaps or [SendImageToVideoFromBitmaps2](#)

It is possible to:

- either invoke SendImageToVideoFromBitmaps from the [OnVideoFromBitmapsNextFrameNeeded](#) event, that occurs periodically to request the next frame,
- either invoke periodically SendImageToVideoFromBitmaps directly. In this 2nd case, SendImageToVideoFromBitmaps must be **called one time just before invoking** StartPreview() or StartRecording().

See [Video clips built on the fly by passing bitmap handles, BMP or JPEG files](#) for more information.

You will find sample code in the MainDemo project included in the package, or in the VideoFromBitmapHandles or VideoFromBMPorJPEGfiles demos.

### See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps2](#) [VideoFromImages](#) [BitmapsSortedByVideoFromImages](#) [RepeatIndefinitely](#) [VideoFromImages](#) [SourceDirectory](#) [VideoFromImages](#) [TemporaryFile](#)

---

Created with the Standard Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

---

## SendImageToVideoFromBitmaps2

<b>TVideoGrabber.SendImageToVideoFromBitmaps2</b>	<a href="#">Prev</a>	<a href="#">Next</a>
<a href="#">TVideoGrabber</a> <a href="#">Methods</a>		

Used to pass a pointer to the bitmap bits of the next image to pass to VideoFromJPGsOrBitmaps

### Declaration

```
function SendImageToVideoFromBitmaps2 (pBtmapInfo: LongInt; pBitmapBits: LongInt; EndOfData: Boolean): Boolean;
```

```
bool __fastcall SendImageToVideoFromBitmaps2(Windows::PBitmapInfo pBtmapInfo, System::PByte pBitmapBits, bool EndOfData);
```

Function SendImageToVideoFromBitmaps2 (pBtmapInfo as Long, pBitmapBits as Long, EndOfData as Boolean) as Boolean

### Description

Passes the next image to VideoFromJPGsOrBitmaps as a pointer to the bitmap bits of the next image, when building a clip from bitmaps.

Works the same way as [SendImageToVideoFromBitmaps](#).

### See Also

[OnBitmapsLoadingProgress](#) [SendImageToVideoFromBitmaps](#) [VideoFromImages](#) [BitmapsSortedByVideoFromImages](#) [RepeatIndefinitely](#) [VideoFromImages](#) [SourceDirectory](#)

[VideoFromImages\\_TemporaryFile](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## SendIPCameraCommand

### TVideoGrabber.SendIPCameraCommand

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sends an HTTP command to the IP camera.

#### Declaration

**function** SendIPCameraCommand (IPCameraCommand: **String**): Boolean;

**bool** \_\_fastcall SendIPCameraCommand(wchar\_t \*IPCameraCommand);

function SendIPCameraCommand(IPCameraCommand as String) as Boolean

#### Description

Used to send an HTTP command to the IP camera.

The command string must be an HTTP URL. The syntax depends of the manufacturer.

E.g. to send PTZ commands to an Axis IP Camera with PTZ control:

SendIPCameraCommand ("http://x.x.x.x/axis-cgi/com/ptz.cgi?move=home")

SendIPCameraCommand ("http://x.x.x.x/axis-cgi/com/ptz.cgi?move=up")

etc...

*(replace x.x.x.x by the IP address of the camera)*

The list of the syntax for Axis IP cameras can be found here:

[http://www.axis.com/techsup/cam\\_servers/dev/cam\\_http\\_api\\_index.php](http://www.axis.com/techsup/cam_servers/dev/cam_http_api_index.php)

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras\\_IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Experience the Power and Ease of Use of a Help Authoring Tool](#)

## SetAudioRendererAdditional

### TVideoGrabber.SetAudioRendererAdditional

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Allows to use an additional audio renderer

#### Declaration

procedure SetAudioRendererAdditional (Value: LongInt);

void \_\_fastcall SetAudioRendererAdditional(int Value);

## Description

Allows to render the audio to a second audio renderer, additionally to the first one.

Set AudioRendererAdditional = [AudioRendererIndex](#) ("... name of the audio renderer to use...") among the audio renderers listed in the [AudioRenderers](#) list.

Note: be sure to specify another index that the index of the default audio renderer (the default audio renderer has the index 0).

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## SetAuthentication

### TVideoGrabber.SetAuthentication

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the username and password required to connect to a publishing point, a streaming URL or an IP camera

#### Declaration

**procedure** SetAuthentication (AuthenticationType: TAuthenticationType; UserName: **String**; Password: **String**);

**void \_\_fastcall** SetAuthentication(TAuthenticationType TAuthenticationType; wchar\_t \*UserName; wchar\_t \*Password);

#### Description

Invoke SetAuthentication to set the username and password that will be required soon.

E.g.:

**- to set the username and password for a publishing point on a windows media server:**

```
VideoGrabber.NetworkStreaming = ns_ASFStreamingToPublishingPoint
VideoGrabber.ASFMediaServerPublishingPoint = "http://...."
VideoGrabber.SetAuthentication (at_PublishingPoint, "MyPubpointUser", "MyPubpointPassword");
VideoGrabber.StartPreview
```

**- to set the username and password for an IP camera:**

```
VideoGrabber.VideoSource = vs_IPCamera
VideoGrabber.IPCameraURL = "http://..."
VideoGrabber.SetAuthentication (at_IPCamera, "MyIPCamUser", "MyIPCamPassword");
VideoGrabber.StartPreview
```

**- to set the username and password for a streaming URL:**

```
VideoGrabber.VideoSource = vs_VideoFileOrUrl
VideoGrabber.VideoSource_FileOrUrl = "http://.....asf"
VideoGrabber.SetAuthentication (at_StreamingUrl, "MyURLUsername", "MyURLPassword");
VideoGrabber.StartPreview
```

Created with the Standard Edition of HelpNDoc: [Free Qt Help documentation generator](#)

## SetAVIMuxConfig

### TVideoGrabber.SetAVIMuxConfig

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Configures the AVI mux

#### Declaration

**procedure** SetAVIMuxConfig(AVIMuxSetting: TAVIMuxConfig; Value: LongInt);

**void** \_\_fastcall SetAVIMuxConfig(TAVIMuxConfig AVIMuxSetting, **int** Value)

Sub SetAVIMuxConfig (AVIMuxSetting as TAVIMuxConfig, Value as Long)

#### Description

Used to adjust non-default values of the AVI multiplexer.

#### avmx\_SetInterleavingMode

adjusts the interleaving mode:

0 = INTERLEAVE\_NONE

1 = INTERLEAVE\_CAPTURE

2 = INTERLEAVE\_FULL

3 = INTERLEAVE\_NONE\_BUFFERED

#### avmx\_SetInterleave

adjusts the interleave expressed in 100ns units

"The default value for prlInterleave is 1000 milliseconds; however, you can adjust this. The smaller the number, the larger the file."

#### avmx\_SetPreroll:

adjusts the audio preroll

"An audio preroll of 750 milliseconds is recommended when authoring a file for distribution."

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer](#)

## SetCameraControl

### TVideoGrabber.SetCameraControl

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a [TCameraControl](#) setting.

#### Declaration

**function** SetCameraControl(Setting: TCameraControl; SetAuto: Boolean; SetDefault: Boolean; SetValue: LongInt): Boolean;

**bool** \_\_fastcall SetCameraControl(TCameraControl Setting, bool SetAuto, bool SetDefault, int SetValue)

Function SetCameraControl(Setting As TxCameraControl, SetAuto as Boolean, SetDefault as Boolean, SetValue as Long) As Boolean

#### Description

Used to set a [TCameraControl](#) setting, if available for the current video capture device (test [IsCameraControlSetting](#) for availability).

- if **SetAuto** is true, SetDefault and SetValue are ignored and the setting is switched in "auto" mode
- if **SetAuto** is false and **SetDefault** is true, SetValue is ignored and the setting is set to its default value,
- if **SetAuto** is false and **SetDefault** is false, the **SetValue** value is applied to the setting.

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraExposure](#)

## SetCameraExposure

### TVideoGrabber.SetCameraExposure

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the camera exposure as a double value

#### Declaration

**procedure** SetCameraExposure (Value: Double);

**void** \_\_fastcall SetCameraExposure(**double** Value);

Sub SetCameraExposure (Value as Double)

#### Description

Used to set the camera exposure as a double value, e.g. 1/8 s, 1/16 s, 1/32 s, etc...

#### See Also

[TCameraControl](#) [CameraControlAuto](#) [CameraControlDefault](#) [CameraControlMax](#) [CameraControlMin](#) [CameraControlSettings](#) [CameraControlStep](#) [GetCameraExposure](#) [GetCameraExposureAsString](#) [IsCameraControlAvailable](#) [IsCameraControlSettingAvailable](#) [SendCameraCommand](#) [SetCameraControl](#)

## SetDecryptionKey

### TVideoGrabber.SetDecryptionKey

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Key used for decryption

#### Declaration

**function** SetDecryptionKey (Key: pBYTE; KeyLen: LongInt): Boolean;

**bool** \_\_fastcall SetDecryptionKey(System::UnicodeString Value);

#### Description

Sets the key to decrypt the video/audio during recording

#### See Also

[TEncryptionMethod](#) [Decrypt\\_File](#) [Encrypt\\_File](#) [EncryptionMethod](#) [SetEncryptionKey](#)

## SetDisplayActive

### TVideoGrabber.SetDisplayActive

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables/disables the window

## Declaration

**procedure** SetDisplayActive (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayActive(**int** DisplayIndex, **bool** Value);

## Description

Used to enable/disable the video window.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## SetDisplayAlphaBlendEnabled

### TVideoGrabber.SetDisplayAlphaBlendEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Activates the alpha blending of the video window

## Declaration

**procedure** SetDisplayAlphaBlendEnabled (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayAlphaBlendEnabled(**int** DisplayIndex, **bool** Value);

## Description

Used to activate the alpha blending of the current video window with another video window when it is detached from the control by invoking [SetDisplayEmbedded](#) (index, false)

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## SetDisplayAlphaBlendValue

### TVideoGrabber.SetDisplayAlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Value of the alpha blending of the video window

## Declaration

**procedure** SetDisplayAlphaBlendValue (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayAlphaBlendValue(**int** DisplayIndex, **int** Value);

## Description

Value of the alpha blending of the video window, in the 0..255 range (e.g. 128 = semi-transparent)

**This feature can be used after invoking [SetDisplayEmbedded](#) (index, false) and [SetDisplayAlphaBlendEnabled](#) (index, false)**

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

## SetDisplayAspectRatio

### TVideoGrabber.SetDisplayAspectRatio

[Prev](#)

[Next](#)

## [TVideoGrabber](#) [Methods](#)

Specifies the aspect ratio to use within the video window.

### Declaration

**procedure** SetDisplayAspectRatio (DisplayIndex: LongInt; Value: TAspectRatio);

**void \_\_fastcall** SetDisplayAspectRatio(int DisplayIndex, TAspectRatio Value);

### Description

Used after invoking [SetDisplayAutoSize](#) (index, false) to specify the aspect ratio method that must be used within the video window.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer](#)

## SetDisplayAutoSize

### TVideoGrabber.SetDisplayAutoSize

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Methods](#)

If enabled, the control is resized automatically according to the current video size.

### Declaration

**procedure** SetDisplayAutoSize (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayAutoSize(int DisplayIndex, **bool** Value);

### Description

Specifies whether the control must be resized automatically according to the current video size.

**disabled:** the control size depends of the Width and Height properties, and the video window is stretched inside.

**enabled:** the control size is automatically modified according to [VideoSize](#) or [UseNearestVideoSize](#) .

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Full-featured Documentation generator](#)

## SetDisplayEmbedded

### TVideoGrabber.SetDisplayEmbedded

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Methods](#)

Used to detach/attach the video window from the TVideoGrabber control.

### Declaration

**procedure** SetDisplayEmbedded (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayEmbedded(int DisplayIndex, **bool** Value);

### Description

Used to detach/attach the video window from the TVideoGrabber control.

- **enabled:** the video window is embedded into the TVideoGrabber control,

- **disabled:** the video window is located on the desktop at the positions specified by invoking [SetDisplayLeft](#) and [SetDisplayTop](#), and optionally [SetDisplayMonitor](#) if more than 1 monitor is used.



*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Benefits of a Help Authoring Tool](#)

## SetDisplayFullScreen

### TVideoGrabber.SetDisplayFullScreen

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Displays the preview window in full screen mode.

#### Declaration

**procedure** SetDisplayFullScreen (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayFullScreen(int DisplayIndex, **bool** Value);

#### Description

If enabled, the preview window is displayed in full screen mode when the preview, recording or play back starts.

To exit from the full screen mode disable this property or press the <ESC> key.

In full screen mode all keystrokes are returned by the [OnKeyPress](#) event.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## SetDisplayHeight

### TVideoGrabber.SetDisplayHeight

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the height of the video window, when it is not embedded in the TVideoGrabber control.

#### Declaration

**procedure** SetDisplayHeight (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayHeight(int DisplayIndex, **int** Value);

#### Description

Used to specify the height of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by invoking [SetDisplayLocation](#).

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## SetDisplayLeft

### TVideoGrabber.SetDisplayLeft

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the left position of the video window, when it is not embedded in the TVideoGrabber control.

#### Declaration

**procedure** SetDisplayLeft (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayLeft(**int** DisplayIndex, **int** Value);

#### Description

Specifies the left position of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by invoking [SetDisplayLocation](#).

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

## SetDisplayLocation

### TVideoGrabber.SetDisplayLocation

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the location of the display window

#### Declaration

**function** SetDisplayLocation (DisplayIndex: LongInt; WindowLeft: LongInt; WindowTop: LongInt; WindowWidth: LongInt; WindowHeight: LongInt): Boolean;

**bool \_\_fastcall** SetDisplayLocation(**int** DisplayIndex, **int** WindowLeft, **int** WindowTop, **int** WindowWidth, **int** WindowHeight);

#### Description

Used to set the location and size of the display window when it is not embedded in the control, after invoking SetDisplayEmbedded (index, false)

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

## SetDisplayMonitor

### TVideoGrabber.SetDisplayMonitor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the monitor used to display the video window.

#### Declaration

**procedure** SetDisplayMonitor (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayMonitor(**int** DisplayIndex, **int** Value);

#### Description

Used to specify the monitor used to display the video window.  
The value is in the ( 0..[MonitorsCount](#) -1 ) range.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## SetDisplayMouseMovesWindow

### TVideoGrabber.SetDisplayMouseMovesWindow

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

## ber

Specifies whether the mouse moves the video window or returns mouse events for this window.

### Declaration

**procedure** SetDisplayMouseMovesWindow (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayMouseMovesWindow(**int** DisplayIndex, **bool** Value);

### Description

When the video window is not embedded in the TVideoGrabber control:

- **if enabled:** the mouse moves the video window,
- **if disabled:** the mouse returns mouse events, e.g. to draw graphic objects over the video window.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Transform your help documentation into a stunning website](#)

## SetDisplayPanScanRatio

### TVideoGrabber.SetDisplayPanScanRatio

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Adjust the Pan/Scan ratio.

### Declaration

**procedure** SetDisplayPanScanRatio (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayPanScanRatio(**int** DisplayIndex, **int** Value);

### Description

After invoking [SetDisplayAspectRatio](#) (index, ar\_PanScan), this function can be used to adjust the Pan/Scan ratio (in the 0..100 range, default value 50)

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## SetDisplayParent

### TVideoGrabber.SetDisplayParent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the display parent of the video window

### Declaration

**function** SetDisplayParent (DisplayIndex: LongInt; DisplayParent: HWND): Boolean;

**bool \_\_fastcall** SetDisplayParent(**int** DisplayIndex, HWND DisplayParent);

### Description

Invoke it to set the parent of the video window. The parent parameter must be a HWND window handle.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## SetDisplayStayOnTop

### TVideoGrabber.SetDisplayStayOnTop

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies whether the video window must stay over other windows, when not embedded.

#### Declaration

**procedure** SetDisplayStayOnTop (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayStayOnTop(**int** DisplayIndex, **bool** Value);

#### Description

Specifies whether the video window must stay over other windows, when it is not embedded in the TVideoGrabber control.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## SetDisplayTop

### TVideoGrabber.SetDisplayTop

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the top position of the video window, when it is not embedded in the TVideoGrabber control.

#### Declaration

**procedure** SetDisplayTop (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayTop(**int** DisplayIndex, **int** Value);

#### Description

Specifies the top position of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by invoking [SetDisplayLocation](#).

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Easy EBook and documentation generator](#)

## SetDisplayTransparentColorEnabled

### TVideoGrabber.SetDisplayTransparentColorEnabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Activate the color keying of the current video window

#### Declaration

**procedure** SetDisplayTransparentColorEnabled (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayTransparentColorEnabled(**int** DisplayIndex, **bool** Value);

#### Description

Used to activate the color keying of the current video window with another video window when it is detached from the control by invoking [SetDisplayEmbedded](#) (index, false)

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## SetDisplayTransparentColorValue

### TVideoGrabber.SetDisplayTransparentColorValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Value of the color keying of the video window

#### Declaration

**procedure** SetDisplayTransparentColorValue (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayTransparentColorValue(**int** DisplayIndex, **int** Value);

#### Description

Value of the color keying of the video window, expressed in RGB value.

**This feature can be used after invoking [SetDisplayEmbedded](#) (index, false) and [SetDisplayTransparentColorEnabled](#) (index, false)**

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Transform Your Documentation Workflow with HelpNDoc's Intuitive UI](#)

## SetDisplayVideoPortEnabled

### TVideoGrabber.SetDisplayVideoPortEnabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the video port renderer

#### Declaration

**procedure** SetDisplayVideoPortEnabled (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayVideoPortEnabled(**int** DisplayIndex, **bool** Value);

#### Description

Used to enable the video port renderer, if available

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## SetDisplayVisible

### TVideoGrabber.SetDisplayVisible

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Shows / hide the video window

#### Declaration

**procedure** SetDisplayVisible (DisplayIndex: LongInt; Value: Boolean);

**void \_\_fastcall** SetDisplayVisible(**int** DisplayIndex, **bool** Value);

#### Description

Used to show / hide the video window.  
Enabled by default.

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

## SetDisplayWidth

### TVideoGrabber.SetDisplayWidth

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the width of the video window, when it is not embedded in the TVideoGrabber control.

#### Declaration

**procedure** SetDisplayWidth (DisplayIndex: LongInt; Value: LongInt);

**void \_\_fastcall** SetDisplayWidth(**int** DisplayIndex, **int** Value);

#### Description

Used to specify the width of the video window, when it is not embedded in the TVideoGrabber control.

Note: the left, top, width and height properties can be set at the same time by invoking [SetDisplayLocation](#).

*The window Index specified must be in the 2..8 range (0..1 reserved to Display... and Dual\_Display)*

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## SetEncryptionKey

### TVideoGrabber.SetEncryptionKey

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Key used for encryption

#### Declaration

**function** SetEncryptionKey (Key: pBYTE; KeyLen: LongInt): Boolean;

**bool \_\_fastcall** SetEncryptionKey(System::UnicodeString Value);

#### Description

Sets the key to encrypt the video/audio during recording

#### See Also

[TEncryptionMethod](#) [Decrypt\\_File](#) [Encrypt\\_File](#) [EncryptionMethod](#) [SetDecryptionKey](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## SetFFmpegAudioFilter

### TVideoGrabber.SetFFmpegAudioFilter

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

enables a FFmpeg audio filter

#### Declaration

**function** SetFFmpegFilter (FilterIndex: LongInt; VideoFilterName: **string**; Parameters: **string**): Boolean;

**bool \_\_fastcall** SetFFmpegFilter(System::LongInt FilterIndex, System::UnicodeString VideoFilterName, System::UnicodeString Parameters);

### Description

Used to activate a FFmpeg audio filter.

Note: this feature requires the Datastead RTSP/RTMP/HTTP/ONVIF Source Filter to be installed.

Note: the filter must be an "in place" filter that outputs the same video format and video size than the input format.

Example to perform a voice masking:

```
VideoGrabber.SetFFmpegAudioFilter(0, "afftfilt",
"real='hypot(re,im)':imag='0':win_size=2048:overlap=0.75:win_func=hann");
VideoGrabber.SetFFmpegAudioFilter(1, "asetrate", "52000");
```

### See Also

[MultipurposeEncoder\\_QuickConfigure\\_UDPStreaming\\_H264](#)

[MultipurposeEncoder\\_ReindexClip](#) [SetFFmpegFilter](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## SetFFmpegFilter

### TVideoGrabber.SetFFmpegFilter

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

enables a FFmpeg video filter

### Declaration

**function** SetFFmpegFilter (FilterIndex: LongInt; VideoFilterName: **string**; Parameters: **string**): Boolean;

**bool \_\_fastcall** SetFFmpegFilter(System::LongInt FilterIndex, System::UnicodeString VideoFilterName, System::UnicodeString Parameters);

### Description

Used to activate a FFmpeg video filter.

Note: this feature requires the Datastead RTSP/RTMP/HTTP/ONVIF Source Filter to be installed.

Note: the filter must be an "in place" filter that outputs the same video format and video size than the input format.

Example:

```
VideoGrabber.SetFFmpegFilter (0, 'vignette',
'angle=PI/2:mode=forward:x0=w/2:y0=h/2:dither=0');
VideoGrabber.SetFFmpegFilter (1, 'vflip', '');
```

### See Also

[MultipurposeEncoder\\_QuickConfigure\\_UDPStreaming\\_H264](#)

[MultipurposeEncoder\\_ReindexClip](#) [SetFFmpegAudioFilter](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Publish Your Word Document as an eBook](#)

## SetFrameCaptureBounds

### TVideoGrabber.SetFrameCaptureBounds

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies a frame capture rectangle

### Declaration



**procedure** SetFrameCaptureBounds (LeftPosition: LongInt; TopPosition: LongInt; RightPosition: LongInt; BottomPosition: LongInt);

**void \_\_fastcall** SetFrameCaptureBounds(**int** LeftPosition, **int** TopPosition, **int** RightPosition, **int** BottomPosition)

Sub SetFrameCaptureBounds (LeftPosition as Long, TopPosition as Long, RightPosition as Long, BottomPosition as Long)

### Description

**Used to capture a sub-rectangle of the video frame**

Used to specify a rectangle for the frame capture, allowing to capture only parts of the video frame.

E.g.:

*Button1 action:*

VideoGrabber.StartPreview()

*Button2 action:*

VideoGrabber.SetFrameCaptureBounds (50, 50, 150, 150)

VideoGrabber.CaptureFrameTo (fc\_JPEGFile, "mysquarecapture.jpg")

When starting the preview with Button1, then capturing a frame with Button2, the captured frame will be a square 100x100 image captured at the "x=50 y=50" location on the video frame.

*Note: the normal full frame capture can then be restored by invoking SetFrameCaptureBounds (0, 0, 0, 0)*

### Top-down or left-right capture

Additionally it is possible top-down or left-right the captured frame by just inverting the Left/Right or Top/Bottom values.

E.g.:

VideoGrabber.SetFrameCaptureBounds (50, **150**, 150, **50**) will capture a top-down frame.

**Remark: to retrieve the current size of the video frame use [VideoWidth](#) and [VideoHeight](#)**

### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [StoragePath](#) [WebcamStillCaptureButton](#)

---

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

---

## SetFWCam1394

### TVideoGrabber.SetFWCam1394

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a custom setting of Firewire/GIE camera

### Declaration

**function** SetFWCam1394 (FWCam1394ID: **string**; Value: LongInt): Boolean;

**bool \_\_fastcall** SetFWCam1394(System::wchar\_t \*FWCam1394ID, **int** Value);

## Description

Example in Delphi:

```

procedure TfrmMainForm.Button14Click(Sender: TObject);
var
  Value: LongInt;
  Flags: LongInt;
  Capabilities: DWORD;
  MinValue: LongInt;
  MaxValue: LongInt;
Default: LongInt;
begin
  if VideoGrabber.GetFWCam1394 ('AUTO_EXPOSURE_REFERENCE', Value, Flags,
  Capabilities, MinValue, MaxValue, Default) then begin
  mmoLog.lines.add(inttostr (Value));
  mmoLog.lines.add(inttostr (Flags));
  mmoLog.lines.add(inttostr (Capabilities));
  mmoLog.lines.add(inttostr (MinValue));
  mmoLog.lines.add(inttostr (MaxValue));
  mmoLog.lines.add(inttostr (Default));
  if VideoGrabber.SetFWCam1394 ('AUTO_EXPOSURE_REFERENCE', 2) then begin
  showmessage('new value applied');
  end;
  if VideoGrabber.SetFWCam1394 ('AUTO_EXPOSURE_REFERENCE', 128) then begin
  showmessage('new value applied');
  end;
end;
end;
end;

```

---

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

---

## SetHeaderAttribute

### TVideoGrabber.SetHeaderAttribute

[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the header attributes of AVI / ASF files.

#### Declaration

```
procedure SetHeaderAttribute(HeaderAttribute: THeaderAttribute; Value: String);
```

```
void __fastcall SetHeaderAttribute(THeaderAttribute HeaderAttributeGraphics, wchar_t *Value)
```

```
Sub SetHeaderAttribute(HeaderAttribute as TxHeaderAttribute, Value as String)
```

#### Description

Used to set the header attributes (strings like Author, Description, etc...) to the AVI or ASF files. See [AVI & ASF header attributes](#) for more information.

The allowed values for HeaderAttributes are:

- for both AVI and ASF files (RecordingMethod = rm\_AVI or rm\_ASF)

```

ha_Title
ha_Description
ha_Author
ha_Copyright

```

- for ASF files only (RecordingMethod = rm\_ASF)

ha\_AlbumArtist  
 ha\_AlbumTitle  
 ha\_Composer  
 ha\_ContentDistributor  
 ha\_Director  
 ha\_EncodingTime  
 ha\_Genre  
 ha\_Language  
 ha\_ParentalRating  
 ha\_Producer  
 ha\_Provider  
 ha\_ToolName  
 ha\_ToolVersion  
 ha\_Writer

**- for AVI files only (RecordingMethod = rm\_AVI)**

*(from the Multimedia Programming Interface and Data Specifications 1.0)*

ha\_IARL: Archival Location. Indicates where the subject of the file is archived.

ha\_ICMS: Commissioned. Lists the name of the person or organization that commissioned the subject of the file. For examplePope Julian II.

ha\_ICMT: Comments. Provides general comments about the file or the subject of the file. If the comment is several sentences longend each sentence with a period. Do not include newline characters.

ha\_ICRD: Creation date. Specifies the date the subject of the file was created. List dates in year-month-day formatpadding one-digit months and days with a zero on the left. For example1553-05-03 for May 31553.

ha\_ICRP: Cropped. Describes whether an image has been cropped andif sohow it was cropped. For examplelower right corner.

ha\_IDIM: Dimensions. Specifies the size of the original subject of the file.

ha\_IDPI: Dots Per Inch. Stores dots per inch setting of the digitizer used to produce the filesuch as 300.

ha\_IENG: Engineer. Stores the name of the engineer who worked on the file. If there are multiple engineersseparate the names by a semicolon and a blank. For exampleSmithJohn; AdamsJoe.

ha\_IGNR: Genre. Describes the original worksuch aslandscapeportraitstill lifeetc.

ha\_IKEY: Keywords. Provides a list of keywords that refer to the file or subject of the file. Separate multiple keywords with a semicolon and a blank. For exampleSeattle; aerial view; scenery.

ha\_ILGT: Lightness. Describes the changes in lightness settings on the digitizer required to produce the file. Note that the format of this information depends on hardware used.

ha\_IMED: Medium. Describes the original subject of the filesuch ascomputer imagedrawinglithographand so forth.

ha\_IPLT: Palette Setting. Specifies the number of colors requested when digitizing an imagesuch as 256.

ha\_IPRD: Product. Specifies the name of the title the file was originally intended forsuch as Encyclopedia of Pacific Northwest Geography.

ha\_ISFT: Software. Identifies the name of the software package used to create the filesuch as Microsoft WaveEdit.

ha\_ISHP: Sharpness. Identifies the changes in sharpness for the digitizer required to produce the file (the format depends on the hardware used).

ha\_ISRc:/tab Source. Identifies the name of the person or organization who supplied the original subject of the file. For exampleTrey Research.

ha\_ISRF: Source Form. Identifies the original form of the material that was digitizedsuch as slidepapermapand so forth. This is not necessarily the same as IMED.

ha\_ITCH: Technician. Identifies the technician who digitized the subject file. For exampleSmithJohn.

**See Also**

[THeaderAttribute](#) [ClearHeaderAttributes](#)

## SetImageOverlay\_AlphaBlend

### TVideoGrabber.SetImageOverlay\_AlphaBlend

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the alpha blending for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_AlphaBlend (**Index**: LongInt; **Value**: Boolean);

**void** \_\_fastcall SetImageOverlay\_AlphaBlend(**int** Index, **bool** Value);

#### Description

Enables the alpha blending for the image overlay of the specified index

The alpha blending value must be specified with [SetImageOverlay\\_AlphaBlend](#)

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easy Qt Help documentation editor](#)

## SetImageOverlay\_AlphaBlendValue

### TVideoGrabber.SetImageOverlay\_AlphaBlendValue

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the the alpha blending value for the specified image overlay

**Declaration**

**procedure** SetImageOverlay\_AlphaBlendValue (**Index**: LongInt; **Value** :LongInt);

**void** \_\_fastcall SetImageOverlay\_AlphaBlendValue(**int** Index, **int** Value);

**Description**

Sets the alpha blending value for the image overlay of the specified index

The alpha blending must be enabled by [SetImageOverlay\\_AlphaBlend](#)

Look at the [Image Overlays](#) chapter for more information.

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#)  
[ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#)  
[ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#)  
[ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#)  
[SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#)  
[SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#)  
[SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#)  
[SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#)  
[SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#)  
[SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Enhance Your Documentation with HelpNDoc's Advanced Project Analyzer](#)

**SetImageOverlay\_Attributes****TVideoGrabber.SetImageOverlay\_Attributes**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the location, size and transparency for the current image overlay selected by [ImageOverlaySelector](#)

**Declaration**

**procedure** SetImageOverlayAttributes (LeftLocation: LongInt; TopLocation: LongInt; StretchedWidth: LongInt; StretchedHeight: LongInt; Transparent: Boolean; UseTransparentColor: Boolean; TransparentColorValue: LongInt; AlphaBlend: Boolean; AlphaBlendValue: LongInt);

**void** \_\_fastcall SetImageOverlayAttributes(**int** LeftLocation, **int** TopLocation, **int** StretchedWidth, **int** StretchedHeight, **bool** Transparent, **bool** UseTransparentColor, **TColor** TransparentColorValue, **bool** AlphaBlend, **int** AlphaBlendValue)

Sub SetImageOverlayAttributes LeftLocation As Long, TopLocation As Long, bmWidth As Long, bmHeight As Long, Transparent As Boolean, UseTransparentColor As Boolean, TransparentColorValue As Long, AlphaBlend As Boolean, AlphaBlendValue As Long)

### Description

**Used to Specify the location, size and transparency of the image overlayed over the video frames**

*Note: see the [Image overlays](#) chapter for global information about this feature.*

### Location

The image will be drawn at the LeftLocation, TopLocation position over the video frames (expressed in pixels).

### Size

- if StretchedWidth and StretchedHeight are > 0, the image is resized to these values
- if StretchedWidth = 0 and StretchedHeight = 0, the image size is used

### Transparency

- if the **Transparent** parameter is true, the image background is not drawn.
- if the **Transparent** parameter is true and the **UseTransparentColor** parameter is true, the **TransparentColorValue** value is used for transparency.

### Alpha blending

If the **AlphaBlend** parameter is true, a progressive bitmap transparency is applied, depending of the **AlphaBlendValue**, in a 0..255 range.

It is not possible to perform alphablending when the Transparent parameter is true.

#### Important:

if the bitmap color format is not 32 bits, it will be converted into a 32 bit format. In other words, for better performance pass a bitmap already in 32 bits color format.

### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)



[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## SetImageOverlay\_Attributes2

### TVideoGrabber.SetImageOverlay\_Attributes2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the location, size and transparency for the image overlay specified by its index

#### Declaration

**procedure** SetImageOverlayAttributes2 (Index: LongInt; LeftLocation: LongInt; TopLocation: LongInt; StretchedWidth: LongInt; StretchedHeight: LongInt; Transparent: Boolean; UseTransparentColor: Boolean; TransparentColorValue: LongInt; AlphaBlend: Boolean; AlphaBlendValue: LongInt);

**void \_\_fastcall** SetImageOverlayAttributes2(**int** Index, **int** LeftLocation, **int** TopLocation, **int** StretchedWidth, **int** StretchedHeight, **bool** Transparent, **bool** UseTransparentColor, **int** TransparentColorValue, **bool** AlphaBlend, **int** AlphaBlendValue)

Sub SetImageOverlayAttributes2 (Index as Long, LeftLocation As Long, TopLocation As Long, bmWidth As Long, bmHeight As Long, Transparent As Boolean, UseTransparentColor As Boolean, TransparentColorValue As Long, AlphaBlend As Boolean, AlphaBlendValue As Long)

#### Description

Same as [SetImageOverlay\\_Attribute](#), but with the overlay Index as first parameter, so it is not necessary to pre-select it with ImageOverlaySelector.

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## SetImageOverlay\_ChromaKey

### TVideoGrabber.SetImageOverlay\_ChromaKey

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables/disables the chroma key feature for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_ChromaKey (Index: LongInt; Value :Boolean);

**void \_\_fastcall** SetImageOverlay\_ChromaKey(**int** Index, **bool** Value);

#### Description

Used to enable or disable the chroma key feature for the image overlay of the specified index

See the [Chroma Key](#) chapter.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)



[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#)  
[SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#)  
[SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#)  
[SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#)  
[SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#)  
[SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#)  
[SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## SetImageOverlay\_ChromaKeyLeewayPercent

### TVideoGrabber.SetImageOverlay\_ChromaKeyLeewayPercent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the percentage of leeway of the chroma key for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_ChromaKeyLeewayPercent (**Index**: LongInt; **Value** :LongInt);

**void \_\_fastcall** SetImageOverlay\_ChromaKeyLeewayPercent(**int** Index, **int** Value);

#### Description

Used to set the percentage of leeway of the chroma key for the image overlay of the specified index

See the [Chroma Key](#) chapter.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)

[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Documentation Capabilities with  
 HelpNDoc's User-Friendly UI

## SetImageOverlay\_ChromaKeyRGBColor

### TVideoGrabber.SetImageOverlay\_ChromaKeyRGBColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the RGB color used as chroma key for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_ChromaKeyRGBColor (**Index**: LongInt; **Value** :LongInt);

**void** \_\_fastcall SetImageOverlay\_ChromaKeyRGBColor(int Index, int Value);

#### Description

Sets the RGB color used as chroma key for the image overlay of the specified index.

See the [Chroma Key](#) chapter.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#)  
[ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#)  
[ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#)  
[ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#)  
[SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#)  
[SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#)  
[SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#)  
[SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#)  
[SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#)  
[SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#)  
[SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#)  
[SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)

[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## SetImageOverlay\_Enabled

### TVideoGrabber.SetImageOverlay\_Enabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables / disables the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_Enabled (**Index**: LongInt; **Value**: Boolean);

**void** \_\_fastcall SetImageOverlay\_Enabled(int Index, **bool** Value);

#### Description

Used to enable disable the image overlay with specified index.

*Note: see the [Image overlays](#) chapter for global information about this feature.*

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEONHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## SetImageOverlay\_Height

### TVideoGrabber.SetImageOverlay\_Height

[Prev](#)

[Next](#)

## **TVideoGrabber** **Methods**

Sets the height for the specified image overlay

### **Declaration**

**procedure** SetImageOverlay\_Height (**Index**: LongInt; Value :LongInt);

**void \_\_fastcall** SetImageOverlay\_Height(**int** Index, **int** Value);

### **Description**

Sets the height for the image overlay of the specified index.

Set -1 (default) to use the original height of the loaded image.

Look at the [Image Overlays](#) chapter for more information.

### **See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## **SetImageOverlay\_LeftLocation**

### **TVideoGrabber.SetImageOverlay\_LeftLocation**

[Prev](#)

[Next](#)

## **TVideoGrabber** **Methods**

Sets the left location for the specified image overlay

### **Declaration**

**procedure** SetImageOverlay\_LeftLocation (**Index**: LongInt; Value :LongInt);



**void \_\_fastcall** SetImageOverlay\_LeftLocation(**int** Index, **int** Value);

### Description

Set the left location (on the video frames) for the image overlay of the specified index

The value must be between 0 and the current video width.

Look at the [Image Overlays](#) chapter for more information.

### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with a Help Authoring Tool](#)

## SetImageOverlay\_RotationAngle

### TVideoGrabber.SetImageOverlay\_RotationAngle

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a rotation angle for the specified image overlay

### Declaration

**procedure** SetImageOverlay\_RotationAngle (**Index**: LongInt; **Value**: Double);

**void \_\_fastcall** SetImageOverlay\_RotationAngle(**int** Index, **double** Value);

### Description

Used to specify a rotation angle for the image overlay of the specified index

By default the rotation angle is 0.0, it can be any value between 0.0 and 360.0

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#)  
[SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Keep Your Sensitive PDFs Safe with These Easy Security Measures

**SetImageOverlay\_StretchToVideoSize****TVideoGrabber.SetImageOverlay\_StretchToVideoSize**[Prev](#)[Next](#)

**TVideoGrabber** [Methods](#)

Stretches the specified image overlay to the size of the video frames

**Declaration**

**procedure** SetImageOverlay\_StretchToVideoSize (**Index**: LongInt; **Value** :Boolean);

**void** \_\_fastcall SetImageOverlay\_StretchToVideoSize(int Index, **bool** Value);

**Description**

When enabled this property stretches the image overlay to the video size (in this case the image overlay location and size settings are ignored)

See [Graphic overlays](#)

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#) [OnFrameOverlayUsingDC](#)

[OnFrameOverlayUsingDIB](#)
[OnFrameOverlayUsingVIDEOHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)
[OnMouseWheel](#)
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay](#)
[AlphaBlend](#)
[SetImageOverlay](#)
[AlphaBlendValue](#)
[SetImageOverlay](#)
[ChromaKey](#)
[SetImageOverlay](#)
[ChromaKeyLeewayPercent](#)
[SetImageOverlay](#)
[ChromaKeyRGBColor](#)
[SetImageOverlay](#)
[Enabled](#)
[SetImageOverlay](#)
[Height](#)
[SetImageOverlay](#)
[LeftLocation](#)
[SetImageOverlay](#)
[RotationAngle](#)
[SetImageOverlay](#)
[TargetDisplay](#)
[SetImageOverlay](#)
[TopLocation](#)
[SetImageOverlay](#)
[Transparent](#)
[SetImageOverlay](#)
[TransparentColorValue](#)
[SetImageOverlay](#)
[UseTransparentColor](#)
[SetImageOverlay](#)
[Width](#)
[SetImageOverlayFromBMPFile](#)
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)
[SetTextOverlay](#)
[Align](#)
[SetTextOverlay](#)
[BkColor](#)
[SetTextOverlay](#)
[CustomVar](#)
[SetTextOverlay](#)
[Enabled](#)
[SetTextOverlay](#)
[Font](#)
[SetTextOverlay](#)
[FontColor](#)
[SetTextOverlay](#)
[GradientColor](#)
[SetTextOverlay](#)
[GradientMode](#)
[SetTextOverlay](#)
[HighResFont](#)
[SetTextOverlay](#)
[Left](#)
[SetTextOverlay](#)
[Right](#)
[SetTextOverlay](#)
[Scrolling](#)
[SetTextOverlay](#)
[ScrollingSpeed](#)
[SetTextOverlay](#)
[Shadow](#)
[SetTextOverlay](#)
[ShadowColor](#)
[SetTextOverlay](#)
[ShadowDirection](#)
[SetTextOverlay](#)
[String](#)
[SetTextOverlay](#)
[TargetDisplay](#)
[SetTextOverlay](#)
[Top](#)
[SetTextOverlay](#)
[Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay](#)
[Align](#)
[TextOverlay](#)
[BkColor](#)
[TextOverlay](#)
[CreateCustomFont](#)
[TextOverlay](#)
[CreateCustomFont2](#)
[TextOverlay](#)
[Enabled](#)
[TextOverlay](#)
[Font](#)
[TextOverlay](#)
[FontColor](#)
[TextOverlay](#)
[Left](#)
[TextOverlay](#)
[Right](#)
[TextOverlay](#)
[Scrolling](#)
[TextOverlay](#)
[ScrollingSpeed](#)
[TextOverlay](#)
[Selector](#)
[TextOverlay](#)
[Shadow](#)
[TextOverlay](#)
[ShadowColor](#)
[TextOverlay](#)
[ShadowDirection](#)
[TextOverlay](#)
[String](#)
[TextOverlay](#)
[Top](#)
[TextOverlay](#)
[Transparent](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Revolutionize Your CHM Help File Output with HelpNDoc

## SetImageOverlay\_TargetDisplay

### TVideoGrabber.SetImageOverlay\_TargetDisplay

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Specifies the location of the overlay (frame or video window)

#### Declaration

**procedure** SetImageOverlay\_TargetDisplay (**Index**: LongInt; **Value**: LongInt);

**void** \_\_fastcall SetImageOverlay\_TargetDisplay(**int** Index, **int** Value);

#### Description

Specifies if the overlay will be applied to the video frame or to the video window

See

#### See Also

[Image overlays](#)
[TCardinalDirection](#)
[TOnFrameOverlayUsingDC](#)
[TOnFrameOverlayUsingDIB](#)
[TTextOverlayAlign](#)
[AdjustOverlayAspectRatio](#)
[DrawBitmapOverFrame](#)
[GetFrameInfoString](#)
[GetPixelsDistance](#)
[GetRGBPixelAt](#)
[ImageOverlay](#)
[AlphaBlend](#)
[ImageOverlay](#)
[AlphaBlendValue](#)
[ImageOverlay](#)
[Height](#)
[ImageOverlay](#)
[LeftLocation](#)
[ImageOverlay](#)
[StretchToVideoSize](#)
[ImageOverlay](#)
[TopLocation](#)
[ImageOverlay](#)
[Transparent](#)
[ImageOverlay](#)
[TransparentColorValue](#)
[ImageOverlay](#)
[UseTransparentColor](#)
[ImageOverlay](#)
[VideoAlignment](#)
[ImageOverlay](#)
[Width](#)
[ImageOverlayEnabled](#)
[ImageOverlaySelector](#)
[MouseWheelEventEnabled](#)
[OnFrameBitmap](#)
[OnFrameBitmapEventSynchronon](#)
[OnFrameOverlayUsingDC](#)
[OnFrameOverlayUsingDIB](#)
[OnFrameOverlayUsingVIDEOHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)
[OnMouseWheel](#)
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay](#)
[AlphaBlend](#)
[SetImageOverlay](#)
[AlphaBlendValue](#)
[SetImageOverlay](#)
[ChromaKey](#)
[SetImageOverlay](#)
[ChromaKeyLeewayPercent](#)
[SetImageOverlay](#)
[ChromaKeyRGBColor](#)
[SetImageOverlay](#)
[Enabled](#)
[SetImageOverlay](#)
[Height](#)
[SetImageOverlay](#)
[LeftLocation](#)
[SetImageOverlay](#)
[RotationAngle](#)
[SetImageOverlay](#)
[StretchToVideoSize](#)
[SetImageOverlay](#)
[TopLocation](#)
[SetImageOverlay](#)
[Transparent](#)
[SetImageOverlay](#)
[TransparentColorValue](#)
[SetImageOverlay](#)
[UseTransparentColor](#)
[SetImageOverlay](#)
[Width](#)
[SetImageOverlayFromBMPFile](#)
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)



[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Free help authoring tool

## SetImageOverlay\_TopLocation

### TVideoGrabber.SetImageOverlay\_TopLocation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the top location for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_TopLocation (**Index**: LongInt; **Value** :LongInt);

**void \_\_fastcall** SetImageOverlay\_TopLocation(**int** Index, **int** Value);

#### Description

Set the top location (on the video frames) for the image overlay of the specified index

The value must be between 0 and the current video height.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#)  
[ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#)  
[ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#)  
[ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)

[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## SetImageOverlay\_Transparent

### TVideoGrabber.SetImageOverlay\_Transparent

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the transparency for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_Transparent (**Index**: LongInt; **Value** :Boolean);

**void** \_\_fastcall SetImageOverlay\_Transparent(**int** Index, **bool** Value);

#### Description

Used to enable the transparency for the image overlay of the specified index

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## SetImageOverlay\_TransparentColorValue

### TVideoGrabber.SetImageOverlay\_TransparentColorValue

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Sets the transparency color for the specified image overlay

#### Declaration

**procedure** SetImageOverlay\_TransparentColorValue (**Index**: LongInt; **Value** :LongInt);

**void \_\_fastcall** SetImageOverlay\_TransparentColorValue(**int** Index, **int** Value);

#### Description

Used to set the value of the color used for transparency when [SetImageOverlay\\_Transparent](#) and [SetImageOverlay\\_UseTransparentColor](#) have been enabled.

Look at the [Image Overlays](#) chapter for more information.

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

## SetImageOverlay\_UseTransparentColor

### TVideoGrabber.SetImageOverlay\_UseTransparentColor

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Enables the use of a transparency color for the specified image overlay

**Declaration**

**procedure** SetImageOverlay\_UseTransparentColor (**Index**: LongInt; Value :Boolean);

**void \_\_fastcall** SetImageOverlay\_UseTransparentColor(**int** Index, **bool** Value);

**Description**

Used to activate the transparent color for the image overlay of the specified index.

The overlay transparency must be enabled with [SetImageOverlay\\_Transparent](#)

If enabled, the color must be specified by [SetImageOverlay\\_TransparentColorValue](#).

If disabled, the default color transparency (background color) of the image will be used.

Look at the [Image Overlays](#) chapter for more information.

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchrone](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

**SetImageOverlay\_Width****TVideoGrabber.SetImageOverlay\_Width**
[Prev](#)
[Next](#)

**TVideoGrabber** [Methods](#)

Sets the width for the specified image overlay

**Declaration**

**procedure** SetImageOverlay\_Width (**Index**: LongInt; Value :LongInt);

**void \_\_fastcall** SetImageOverlay\_Width(**int** Index, **int** Value);

## Description

Sets the width for the image overlay of the specified index.

Set -1 (default) to use the original width of the loaded image.

Look at the [Image Overlays](#) chapter for more information.

## See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## SetImageOverlayFromBMPFile

### TVideoGrabber.SetImageOverlayFromBMPFile

[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Deprecated, use [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

## Declaration

**function** SetImageOverlayFromBMPFile(FileName: **string**): Boolean;

**bool** \_\_fastcall SetImageOverlayFromBMPFile(wchar\_t \*FileName)

Function SetImageOverlayFromBMPFile(FileName As String) As Boolean

## Description

Deprecated, kept for backward compatibility, see [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

## See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)



[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronon](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromHBitmap](#)  
[SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#)  
[SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#)  
[SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#)  
[ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

## SetImageOverlayFromBMPFile2

### TVideoGrabber.SetImageOverlayFromBMPFile2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Deprecated, use [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

#### Declaration

**function** SetImageOverlayFromBMPFile2(**Index**: LongInt; **FileName**: **string**): Boolean;

**bool** \_\_fastcall SetImageOverlayFromBMPFile2(**int** Index, wchar\_t \*FileName)

Function SetImageOverlayFromBMPFile2 (Index as Long, FileName As String) As Boolean

#### Description

Deprecated, kept for backward compatibility, see [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## SetImageOverlayFromHBitmap

### TVideoGrabber.SetImageOverlayFromHBitmap

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the current image overlay from a bitmap handle.

**Declaration**

**function** SetImageOverlayFromHBitmap(Bitmap: HBITMAP): Boolean;

**bool** \_\_fastcall SetImageOverlayFromHBitmap(HBITMAP Bitmap)

Function SetImageOverlayFromHBitmap(FileName As String) As Boolean

**Description**

Used to load the image overlay currently selected by [ImageOverlaySelector](#) from a bitmap handle.

*Note: see the [Image overlays](#) chapter for global information about this feature.*

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

**SetImageOverlayFromHBitmap2****TVideoGrabber.SetImageOverlayFromHBitmap2**
[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the specified image overlay from a bitmap handle.

**Declaration**

**function** SetImageOverlayFromHBitmap2(Index: LongInt; Bitmap: HBITMAP): Boolean;

**bool** \_\_fastcall SetImageOverlayFromHBitmap2(int Index, HBITMAP Bitmap)

Function SetImageOverlayFromHBitmap2(Index as Long, FileName As String) As Boolean



## Description

Used to load the image overlay specified by the index parameter from the specified bitmap handle.

*Note: see the [Image overlays](#) chapter for global information about this feature.*

## See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#) [ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#) [ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Experience the Power and Ease of Use of HelpNDoc for CHM Help File Generation

## SetImageOverlayFromImageFile

### TVideoGrabber.SetImageOverlayFromImageFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the current image overlay from a file

## Declaration

**function** SetImageOverlayFromImageFile (FileName: **string**): Boolean;

**bool** \_\_fastcall SetImageOverlayFromImageFile(wchar\_t \*FileName);

Function SetImageOverlayFromImageFile(FileName as String) as Bool

## Description

Used to load the image overlay currently selected by [ImageOverlaySelector](#) from the specified file

Various image formats are supported.

See the [Image overlays](#) chapter for more information about overlays.

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile2](#)  
[SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#)  
[SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#)  
[ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

**SetImageOverlayFromImageFile2****TVideoGrabber.SetImageOverlayFromImageFile2**[Prev](#)[Next](#)

**TVideoGrabber** [Methods](#)

Loads the image overlay specified by the Index parameter from a file

**Declaration**

function SetImageOverlayFromImageFile2 (Index: LongInt; FileName: string): Boolean;

**bool** \_\_fastcall SetImageOverlayFromImageFile2 (int Index, wchar\_t \*FileName);

Function SetImageOverlayFromImageFile2 (Index as Long, FileName as String) as Bool

**Description**

Used to load the image overlay currently selected by the Index parameter from the specified file.

Various image formats are supported.

See the [Image overlays](#) chapter for more information about overlays.

**See Also**

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)

[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#)  
[SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#)  
[ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Easy Qt Help documentation editor

## SetImageOverlayFromJPEGFile

### TVideoGrabber.SetImageOverlayFromJPEGFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Deprecated, use [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

#### Declaration

**function** SetImageOverlayFromJPEGFile(Bitmap: HBITMAP): Boolean;

**bool** \_\_fastcall SetImageOverlayFromJPEGFile(wchar\_t \*FileName)

Function SetImageOverlayFromJPEGFile(FileName As String) As Boolean

#### Description

Deprecated, kept for backward compatibility, see [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)

[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## SetImageOverlayFromJPEGFile2

### TVideoGrabber.SetImageOverlayFromJPEGFile2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Deprecated, use [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

#### Declaration

**function** SetImageOverlayFromJPEGFile(**Index**: LongInt; **Bitmap**: HBITMAP): Boolean;

**bool** \_\_fastcall SetImageOverlayFromJPEGFile(int Index, wchar\_t \*FileName)

Function SetImageOverlayFromJPEGFile(Index as Long, FileName As String) As Boolean

#### Description

Deprecated, kept for backward compatibility, see [SetImageOverlayFromImageFile](#) or [SetImageOverlayFromImageFile2](#)

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## SetImageOverlayFromTBitmap

### TVideoGrabber.SetImageOverlayFromTBitmap

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the current image overlay from a TBitmap (available only in the Delphi and C++Builder versions)

#### Declaration

**function** SetImageOverlayFromTBitmap(**Bitmap**: TBitmap): Boolean;

**bool** \_\_fastcall SetImageOverlayFromTBitmap(TBitmap \*Bitmap)

#### Description

Used to load the image overlay currently selected by [ImageOverlaySelector](#) from a TBitmap component (available only in the Delphi and C++Builder versions)

*Note: see the [Image overlays](#) chapter for global information about this feature.*

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#) [ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)



[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap2](#)  
[SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#)  
[ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## SetImageOverlayFromTBitmap2

### TVideoGrabber.SetImageOverlayFromTBitmap2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the specified image overlay from a TBitmap object (Dephi and C++Builder only)

#### Declaration

**function** SetImageOverlayFromTBitmap2(Index: LongInt; Bitmap: TBitmap): Boolean;

**bool** \_\_fastcall SetImageOverlayFromTBitmap2(int Index, TBitmap \*Bitmap)

#### Description

Used to load the image overlay specified by the index parameter from the specified TBitmap object (Dephi and C++Builder only)

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)

[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#)  
[ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

## SetImageOverlayFromTImage

### TVideoGrabber.SetImageOverlayFromTImage

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the current image overlay from a TImage component (available only in the Delphi and C++Builder versions)

#### Declaration

**function** SetImageOverlayFromTImage(Image: TImage): Boolean;

**bool** \_\_fastcall SetImageOverlayFromTImage(TImage \*Image)

#### Description

Used to load the image overlay currently selected by [ImageOverlaySelector](#) from a TImage component (available only in the Delphi and C++Builder versions)

*Note: see the [Image overlays](#) chapter for global information about this feature.*

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#)  
[TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#)  
[GetRGBPixelAt](#) [ImageOverlay](#) [AlphaBlend](#) [ImageOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [Height](#)  
[ImageOverlay](#) [LeftLocation](#) [ImageOverlay](#) [StretchToVideoSize](#) [ImageOverlay](#) [TopLocation](#)  
[ImageOverlay](#) [Transparent](#) [ImageOverlay](#) [TransparentColorValue](#) [ImageOverlay](#) [UseTransparentColor](#)  
[ImageOverlay](#) [VideoAlignment](#) [ImageOverlay](#) [Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#)  
[SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#)  
[SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#)  
[SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#)  
[SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#)  
[SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#)  
[SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#)  
[TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#)

[TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## SetImageOverlayFromTImage2

### TVideoGrabber.SetImageOverlayFromTImage2

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Loads the specified image overlay from a TImage object (Dephi and C++Builder only)

#### Declaration

**function** SetImageOverlayFromTImage2(Image: TImage): Boolean;

**bool** \_\_fastcall SetImageOverlayFromTImage2(TImage \*Image)

#### Description

Used to load the image overlay specified by the index parameter from the specified TImage object (Dephi and C++Builder only)

#### See Also

[Image overlays](#) [TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetPixelsDistance](#) [GetRGBPixelAt](#) [ImageOverlay\\_AlphaBlend](#) [ImageOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_Height](#) [ImageOverlay\\_LeftLocation](#) [ImageOverlay\\_StretchToVideoSize](#) [ImageOverlay\\_TopLocation](#) [ImageOverlay\\_Transparent](#) [ImageOverlay\\_TransparentColorValue](#) [ImageOverlay\\_UseTransparentColor](#) [ImageOverlay\\_VideoAlignment](#) [ImageOverlay\\_Width](#) [ImageOverlayEnabled](#) [ImageOverlaySelector](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchrone](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## SetIPCameraSetting

### TVideoGrabber.SetIPCameraSetting

[Prev](#)
[Next](#)



## TVideoGrabber [Methods](#)

### Sets camera IP settings

#### Declaration

**function** SetIPCameraSetting (Setting: TIPCameraSetting; Value: LongInt): Boolean;

**bool** \_\_fastcall SetIPCameraSetting(TIPCameraSetting Setting, **int** Value);

function SetIPCameraSetting (Setting as TxIPCameraSetting, Value as Long) as Bool

#### Description

##### Used to adjust IP camera settings

The setting property accepts the following settings:

##### ips\_ConnectionTimeout:

Specifies the connection timeout (when connecting to the IP camera) expressed in milliseconds.

Default value: 10000 (= 10 seconds)

Minimum value: 500 (0.5 second).

If the value is too low you may experience connection failure problems.

E.g. SetIPCameraSetting (ips\_ConnectionTimeout, 5000) sets a connection timeout of 5 seconds

##### ips\_ReceiveTimeout:

Specifies the receive timeout (after the camera preview or recording started) to the IP camera expressed in milliseconds.

Default value: 5000 (= 5 seconds)

Minimum value: 500 (0.5 second).

If the value is too low you may experience unexpected disconnection problems.

E.g. SetIPCameraSetting (ips\_ReceiveTimeout, 1000) sets a receive timeout of 1 second

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#)

Created with the Standard Edition of HelpNDoc: [Free Qt Help documentation generator](#)

### SetLocation

## TVideoGrabber.SetLocation

[Prev](#)
[Next](#)

## TVideoGrabber [Methods](#)

Set the location of the video window within the parent control

#### Declaration

n/a

**void** SetLocation (**int** ILeft, **int** ITop, **int** IWidth, **int** IHeight);

#### Description

Used to set the location of the video window within the parent control

(not available in Delphi / CppBuilder, the component resizes automatically when the parent control is resized)

E.g.:

```
CMFCFormDlg::CMFCFormDlg(CWnd* pParent /*=nullptr*/)
: CDialogEx(IDD_MFCFORM_DIALOG, pParent)
, m_VideoGrabber(NULL)
m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
m_VideoGrabber = new CVideoGrabber(this);
CMFCFormDlg::~CMFCFormDlg()
delete m_VideoGrabber;
BOOL CMFCFormDlg::OnInitDialog()
CDialogEx::OnInitDialog();
m_VideoGrabber->SetParentWindow(this->GetSafeHwnd());
...
return TRUE; // return TRUE unless you set the focus to a control
void CMFCFormDlg::OnSize(UINT nType, int w, int h)
m_VideoGrabber->SetLocation(0, 0, w, h);
void CMFCFormDlg::OnBnClickedOk()
m_VideoGrabber->SetVideoSource(TVideoSource::vs_ScreenRecording);
m_VideoGrabber->StartPreview();
```

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

## SetLogoFromBMPFile

### TVideoGrabber.SetLogoFromBMPFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Set the video logo from a BMP file.

#### Declaration

**function** SetLogoFromBMPFile(FileName: **string**): Boolean;

**bool** \_\_fastcall SetLogoFromBMPFile(wchar\_t \*FileName)

Function SetLogoFromBMPFile(FileName As String) As Boolean

#### Description

Used to set the logo displayed in the inactive video window, by using the specified BMP file.

#### See Also

[Logo displayed in the video window](#) [LogoDisplayed](#) [LogoLayout](#) [SetLogoFromHBitmap](#)  
[SetLogoFromJPEGFile](#) [SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

## SetLogoFromHBitmap

### TVideoGrabber.SetLogoFromHBitmap

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Set the video logo from a bitmap handle.

#### Declaration

**function** SetLogoFromHBitmap(Bitmap: HBITMAP): Boolean;

**bool** \_\_fastcall SetLogoFromHBitmap(HBITMAP Bitmap)

Function SetLogoFromHBitmap(Bitmap As Long) As Boolean

## Description

Used to set the logo displayed in the inactive video window, by using the specified bitmap handle.

## See Also

[Logo displayed in the video window](#) [LogoDisplayed](#) [LogoLayout](#) [SetLogoFromBMPFile](#)  
[SetLogoFromJPEGFile](#) [SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## SetLogoFromJPEGFile

### TVideoGrabber.SetLogoFromJPEGFile

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Set the video logo from a JPEG file.

## Declaration

**function** SetLogoFromJPEGFile(FileName: **string**): Boolean;

**bool** \_\_fastcall SetLogoFromJPEGFile(wchar\_t \*FileName)

Function SetLogoFromJPEGFile(FileName As String) As Boolean

## Description

Used to set the logo displayed in the inactive video window, by using the specified JPEG file.

## See Also

[Logo displayed in the video window](#) [LogoDisplayed](#) [LogoLayout](#) [SetLogoFromBMPFile](#)  
[SetLogoFromHBitmap](#) [SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## SetLogoFromTBitmap

### TVideoGrabber.SetLogoFromTBitmap

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Set the video logo from a TBitmap component.

## Declaration

**function** SetLogoFromTBitmap(Bitmap: TBitmap): Boolean;

**bool** \_\_fastcall SetLogoFromTBitmap(Graphics::TBitmap \*Bitmap)

n/a

## Description

Used to set the logo displayed in the inactive video window, by using the specified TBitmap component.

## See Also

[Logo displayed in the video window](#) [LogoDisplayed](#) [LogoLayout](#) [SetLogoFromBMPFile](#)  
[SetLogoFromHBitmap](#) [SetLogoFromJPEGFile](#)

Created with the Standard Edition of HelpNDoc: [Free help authoring environment](#)

## SetLogoFromTImage

### TVideoGrabber.SetLogoFromTImage

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Set the video logo from a TImage component (Delphi and C++Builder versions only).

#### Declaration

**function** SetLogoFromImage(Image: TImage): Boolean;

**bool** \_\_fastcall SetLogoFromImage(ExtCtrls::TImage \*Image)

n/a

#### Description

Used to set the logo displayed in the inactive video window, by using the specified TImage component (Delphi and C++Builder versions only).

#### See Also

[Logo displayed in the video window](#) [LogoDisplayed](#) [LogoLayout](#) [SetLogoFromBMPFile](#) [SetLogoFromHBitmap](#) [SetLogoFromJPEGFile](#) [SetLogoFromTBitmap](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## SetMultiplexerFilterByName

### TVideoGrabber.SetMultiplexerFilterByName

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Methods](#)

Forces the use of a MPEG multiplexer

#### Declaration

**procedure** SetMultiplexerFilterByName(MultiplexerFilterName: **string**);

**void** \_\_fastcall SetMultiplexerFilterByName(System::wchar\_t \*MultiplexerFilterName)

Sub SetMultiplexerFilterByName (MultiplexerFilterName as String)

#### Description

When a MPEG video compressor and a MPEG audio compressor are selected, TVideoGrabber tries to use the corresponding multiplexer (from the same manufacturer).

However you can tell TVideoGrabber what multiplexer to use with this function.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## SetParentWindow

### TVideoGrabber.SetParentWindow

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to set a parent window

#### Declaration

**procedure** SetParentWindow(Value: HWND);

**void** \_\_fastcall SetParentWindow(HWND Value)

#### Description

This function is exposed by the Visual Studio C++ versions to let you attach the control to a parent window programmatically.

#### See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index](#) [MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## SetTextOverlay\_Align

### TVideoGrabber.SetTextOverlay\_Align

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets or retrieves the alignment used to draw text over video frames for the specified text overlay

#### Declaration

**procedure** SetTextOverlay\_Align (Index: LongInt; Value: TTextOverlayAlign);

**void** \_\_fastcall SetTextOverlay\_Align(int Index, TTextOverlayAlign Value);

#### Description

Used to set or retrieve the alignment used to draw text over video frames.  
The text will be drawn between [TextOverlay\\_Left](#) and [TextOverlay\\_Right](#) positions.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)

[OnFrameOverlayUsingVIDEOHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)
[OnMouseWheel](#)
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay\\_AlphaBlend](#)
[SetImageOverlay\\_AlphaBlendValue](#)
[SetImageOverlay\\_ChromaKey](#)
[SetImageOverlay\\_ChromaKeyLeewayPercent](#)
[SetImageOverlay\\_ChromaKeyRGBColor](#)
[SetImageOverlay\\_Enabled](#)
[SetImageOverlay\\_Height](#)
[SetImageOverlay\\_LeftLocation](#)
[SetImageOverlay\\_RotationAngle](#)
[SetImageOverlay\\_StretchToVideoSize](#)
[SetImageOverlay\\_TargetDisplay](#)
[SetImageOverlay\\_TopLocation](#)
[SetImageOverlay\\_Transparent](#)
[SetImageOverlay\\_TransparentColorValue](#)
[SetImageOverlay\\_UseTransparentColor](#)
[SetImageOverlay\\_Width](#)
[SetImageOverlayFromBMPFile](#)
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)
[SetTextOverlay\\_AlphaBlend](#)
[SetTextOverlay\\_AlphaBlendValue](#)
[SetTextOverlay\\_BkColor](#)
[SetTextOverlay\\_CustomVar](#)
[SetTextOverlay\\_Enabled](#)
[SetTextOverlay\\_Font](#)
[SetTextOverlay\\_FontColor](#)
[SetTextOverlay\\_GradientColor](#)
[SetTextOverlay\\_GradientMode](#)
[SetTextOverlay\\_HighResFont](#)
[SetTextOverlay\\_Left](#)
[SetTextOverlay\\_Right](#)
[SetTextOverlay\\_Scrolling](#)
[SetTextOverlay\\_ScrollingSpeed](#)
[SetTextOverlay\\_Shadow](#)
[SetTextOverlay\\_ShadowColor](#)
[SetTextOverlay\\_ShadowDirection](#)
[SetTextOverlay\\_String](#)
[SetTextOverlay\\_TargetDisplay](#)
[SetTextOverlay\\_Top](#)
[SetTextOverlay\\_Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay\\_Align](#)
[TextOverlay\\_BkColor](#)
[TextOverlay\\_CreateCustomFont](#)
[TextOverlay\\_CreateCustomFont2](#)
[TextOverlay\\_Enabled](#)
[TextOverlay\\_Font](#)
[TextOverlay\\_FontColor](#)
[TextOverlay\\_Left](#)
[TextOverlay\\_Right](#)
[TextOverlay\\_Scrolling](#)
[TextOverlay\\_ScrollingSpeed](#)
[TextOverlay\\_Selector](#)
[TextOverlay\\_Shadow](#)
[TextOverlay\\_ShadowColor](#)
[TextOverlay\\_ShadowDirection](#)
[TextOverlay\\_String](#)
[TextOverlay\\_Top](#)
[TextOverlay\\_Transparent](#)
[TextOverlay\\_VideoAlignment](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Full-featured multi-format Help generator

## SetTextOverlay\_AlphaBlend

### TVideoGrabber.SetTextOverlay\_AlphaBlend

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the alpha blending for the specified text overlay

#### Declaration

**procedure** SetTextOverlay\_AlphaBlend (**Index**: LongInt; **Value**: Boolean);

**void** \_\_fastcall SetTextOverlay\_AlphaBlend(**int** Index, **bool** Value);

#### Description

Enables the alpha blending for the text overlay of the specified index

The alpha blending value must be specified with [SetTextOverlay\\_AlphaBlend](#)

Look at the [Text Overlays](#) chapter for more information.

#### See Also

[GetTextOverlay\\_AlphaBlend](#)
[GetTextOverlay\\_AlphaBlendValue](#)
[SetTextOverlay\\_Align](#)
[SetTextOverlay\\_AlphaBlendValue](#)
[SetTextOverlay\\_BkColor](#)
[SetTextOverlay\\_CustomVar](#)
[SetTextOverlay\\_Enabled](#)
[SetTextOverlay\\_Font](#)
[SetTextOverlay\\_FontColor](#)
[SetTextOverlay\\_GradientColor](#)
[SetTextOverlay\\_GradientMode](#)
[SetTextOverlay\\_HighResFont](#)
[SetTextOverlay\\_Left](#)
[SetTextOverlay\\_Right](#)
[SetTextOverlay\\_Scrolling](#)
[SetTextOverlay\\_ScrollingSpeed](#)
[SetTextOverlay\\_Shadow](#)
[SetTextOverlay\\_ShadowColor](#)
[SetTextOverlay\\_ShadowDirection](#)
[SetTextOverlay\\_String](#)
[SetTextOverlay\\_TargetDisplay](#)
[SetTextOverlay\\_Top](#)
[SetTextOverlay\\_Transparent](#)
[TextOverlay\\_CreateCustomFont](#)
[TextOverlay\\_CreateCustomFont2](#)
[TextOverlay\\_VideoAlignment](#)

Created with the Standard Edition of HelpNDoc: Produce Kindle eBooks easily

## SetTextOverlay\_AlphaBlendValue

### TVideoGrabber.SetTextOverlay\_AlphaBlendValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)



Sets the the alpha blending value for the specified text overlay

#### Declaration

**procedure** SetTextOverlay\_AlphaBlendValue (**Index**: LongInt; **Value** :LongInt);

**void** \_\_fastcall SetTextOverlay\_AlphaBlendValue(**int** Index, **int** Value);

#### Description

Sets the alpha blending value for the text overlay of the specified index

The alpha blending must be enabled by [SetTextOverlay\\_AlphaBlend](#)

Look at the [Text Overlays](#) chapter for more information.

#### See Also

[GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_Align](#)  
[SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_VideoAlignment](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## SetTextOverlay\_BkColor

### TVideoGrabber.SetTextOverlay\_BkColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets or retrieves the background color used to draw text over frames for the specified text overlay

#### Declaration

**procedure** SetTextOverlay\_BkColor (**Index**: LongInt; **Value**: TColor);

**void** \_\_fastcall SetTextOverlay\_BkColor(**int** Index, **Graphics::TColor** Value);

#### Description

Used to set or retrieve the background color used to draw text over frames.

Useful only if [TextOverlay\\_Transparent](#) is disabled.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#)  
[SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#)



[SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## SetTextOverlay\_CustomVar

### TVideoGrabber.SetTextOverlay\_CustomVar

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a custom variable displayed in the TextOverlay string for the specified text overlay

#### Declaration

**procedure** SetTextOverlay\_CustomVar(TextOverlaySelector: LongInt; VarIndex: LongInt; VarText: **String**);

**void** \_\_fastcall SetTextOverlay\_CustomVar(**int** TextOverlaySelector, **int** VarIndex, wchar\_t \*VarText)

Sub SetTextOverlay\_CustomVar(TextOverlaySelector as Long, VarIndex As Long, VarText As String)

#### Description

Used to set the custom variable displayed in the TextOverlay string.  
Setting custom variables is faster than rewriting the full text overlay string, because it saves TVideoGrabber having to parse it again.

Parameters:

**Index:** index of the overlay

**VarIndex:** index of the custom variable that has been set with [SetTextOverlay\\_String](#) (from 0 to 9)

**VarText:** text of the custom variable

E.g., to initialize the 2 text overlay string, that will use custom variables:

```
...
VideoGrabber.SetTextOverlay_String (0, "my first text uses %custom0%")
VideoGrabber.SetTextOverlay_Left (0, 10)
VideoGrabber.SetTextOverlay_Top (0, 10)
VideoGrabber.SetTextOverlay_Enabled (0, true)

VideoGrabber.SetTextOverlay_String (1, "my second text uses %custom3% and %custom4%")
VideoGrabber.SetTextOverlay_Left (1, 20)
VideoGrabber.SetTextOverlay_Top (1, 100)
VideoGrabber.SetTextOverlay_Enabled (1, true)
VideoGrabber.StartPreview
...
```

then to update the custom variables of the 1st text overlay string:

```
...
VideoGrabber.SetTextOverlayCustomVar (0, 0, "value 0")
...
```

and of the 2nd text overlay string:

```
...
VideoGrabber.SetTextOverlayCustomVar (1, 3, "value 3")
```

```
VideoGrabber.SetTextOverlayCustomVar (1, 4, "value 4")
...
```

See also [SetTextOverlay\\_String](#).

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Step-by-Step Guide: How to Turn Your Word Document into an eBook

## SetTextOverlay\_Enabled

### TVideoGrabber.SetTextOverlay\_Enabled

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables/disables drawing text over video frames.

#### Declaration

procedure SetTextOverlay\_Enabled (Index: LongInt; Value: Boolean);

void \_\_fastcall SetTextOverlay\_Enabled(int Index, bool Value);

#### Description

Used to enable /disable text overlay over video frames.

Look at the [Text overlays](#) chapter.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)

[SetImageOverlay\\_AlphaBlendValue](#)
[SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#)
[SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#)
[SetImageOverlay\\_Height](#)
[SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#)
[SetImageOverlay\\_StretchToVideoSize](#)
[SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#)
[SetImageOverlay\\_Transparent](#)
[SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#)
[SetImageOverlay\\_Width](#)
[SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#)
[SetTextOverlay\\_AlphaBlend](#)
[SetTextOverlay\\_AlphaBlendValue](#)
[SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#)
[SetTextOverlay\\_Font](#)
[SetTextOverlay\\_FontColor](#)
[SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#)
[SetTextOverlay\\_HighResFont](#)
[SetTextOverlay\\_Left](#)
[SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#)
[SetTextOverlay\\_ScrollingSpeed](#)
[SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#)
[SetTextOverlay\\_ShadowDirection](#)
[SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#)
[SetTextOverlay\\_Top](#)
[SetTextOverlay\\_Transparent](#)
[ShapeOverlay](#)  
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay\\_Align](#)
[TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#)
[TextOverlay\\_CreateCustomFont2](#)
[TextOverlay\\_Enabled](#)
[TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#)
[TextOverlay\\_Left](#)
[TextOverlay\\_Right](#)
[TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#)
[TextOverlay\\_Selector](#)
[TextOverlay\\_Shadow](#)
[TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#)
[TextOverlay\\_String](#)
[TextOverlay\\_Top](#)
[TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Powerful and User-Friendly Help Authoring Tool for Markdown Documents

## SetTextOverlay\_Font

### TVideoGrabber.SetTextOverlay\_Font

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets/retrieves the font used to draw text over video frames for the specified text overlay

#### Declaration

procedure SetTextOverlay\_Font (Index: LongInt; Value: TFont);

void \_\_fastcall SetTextOverlay\_Font(int Index, Graphics::TFont\* Value);

#### Description

Used to set / retrieve the font used to draw text over video frames.

In C#.NET, use the ToHfont().ToInt32() function of the Font object. E.g.:

```
Font NewFont = new Font("Courier New", 14);
VideoGrabber1.SetTextOverlay_Font (0, NewFont.ToHfont().ToInt32());
VideoGrabber1.SetTextOverlay_Enabled (0, true);
VideoGrabber1.StartPreview();
```

#### See Also

[TCardinalDirection](#)
[TOnFrameOverlayUsingDC](#)
[TOnFrameOverlayUsingDIB](#)
[TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#)
[DrawBitmapOverFrame](#)
[GetFrameInfoString](#)
[GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#)
[ImageOverlay\\_StretchToVideoSize](#)
[MouseWheelEventEnabled](#)  
[OnFrameBitmap](#)
[OnFrameBitmapEventSynchronise](#)
[OnFrameOverlayUsingDC](#)
[OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)
[OnMouseWheel](#)  
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#)
[SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#)
[SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#)
[SetImageOverlay\\_Height](#)
[SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#)
[SetImageOverlay\\_StretchToVideoSize](#)
[SetImageOverlay\\_TargetDisplay](#)

[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#)  
[SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#)  
[SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#)  
[ShapeOverlay\\_ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Achieve Professional Documentation Results with a Help Authoring Tool

## SetTextOverlay\_FontColor

### TVideoGrabber.SetTextOverlay\_FontColor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to adjust the font color for the specified overlay in the OCX and DLL versions

Note: for Delphi and C++Builder set the font directly with [SetTextOverlay\\_Font](#)

#### Declaration

**procedure** SetTextOverlay\_FontColor (**Index**: LongInt; Value: TColor);

void \_\_fastcall SetTextOverlay\_Font(int Index, DWORD Value);

#### Description

Used to adjust the font color of the specified overlay in the OCX and DLL versions

Note: for Delphi and C++Builder set the font directly with [SetTextOverlay\\_Font](#)

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronize](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)

[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

## SetTextOverlay\_GradientColor

### TVideoGrabber.SetTextOverlay\_GradientColor

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the gradient color of the specified overlay

#### Declaration

**procedure** SetTextOverlay\_GradientColor (**Index**: LongInt; **Value**: TColor);

**void** \_\_fastcall SetTextOverlay\_GradientColor(**int** Index, Graphics::TColor Value);

#### Description

Used to set the gradient color of the specified overlay.

The gradient mode must first be enabled with [SetTextOverlay\\_GradientMode](#)

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)



## SetTextOverlay\_GradientMode

### TVideoGrabber.SetTextOverlay\_GradientMode

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables enable the gradient mode of the text overlays

#### Declaration

**procedure** SetTextOverlay\_GradientMode (**Index**: LongInt; **Value**: TTextOverlayGradientMode);

**void \_\_fastcall** SetTextOverlay\_GradientMode(**int** Index, TTextOverlayGradientMode Value);

#### Description

Enables enable the gradient mode of the text overlays, and select the orientation

See [TTextOverlayGradientMode](#)

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEODR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Keep Your PDFs Safe from Unauthorized Access with These Security Measures

## SetTextOverlay\_HighResFont

### TVideoGrabber.SetTextOverlay\_HighResFont

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the high-resolution font mode

#### Declaration

**procedure** SetTextOverlay\_HighResFont (**Index**: LongInt; **Value**: Boolean);

```
void __fastcall SetTextOverlay_HighResFont(int Index, bool Value);
```

### Description

Select the high-resolution fonts for the text overlays.

*Note that the high resolution fonts are supported in C++, C# and VB.NET, ActiveX, Lazarus and Delphi/C++Builder XE or higher.*

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseMove](#) [OnMouseDown](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

---

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

---

## SetTextOverlay\_Left

### TVideoGrabber.SetTextOverlay\_Left

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Left position (in pixels) where the text will be drawn over video frames for the specified text overlay

### Declaration

procedure SetTextOverlay\_Left (Index: LongInt; Value: LongInt);

```
void __fastcall SetTextOverlay_Left(int Index, int Value);
```

### Description

Used to set or retrieve the left position (in pixels) where the text will be drawn over video frames. The text will be left-padded on this position if [SetTextOverlay\\_Align](#) has been set to `tf_Left`.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)



[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [AlphaBlend](#) [SetTextOverlay](#) [AlphaBlendValue](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TextOverlay](#) [VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Single source CHM, PDF, DOC and HTML Help creation](#)

## SetTextOverlay\_Right

### TVideoGrabber.SetTextOverlay\_Right

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Right position (in pixels) where the text will be drawn over video frames for the specified text overlay

#### Declaration

procedure SetTextOverlay\_Right (Index: LongInt; Value: LongInt);

void \_\_fastcall SetTextOverlay\_Right(int Index, int Value);

#### Description

Used to set or retrieve the right position (in pixels) where the text will be drawn over video frames. The text will be left-padded on this position if [SetTextOverlay\\_Align](#) has been set to tf\_Right.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay](#) [AlphaBlend](#)  
[GetTextOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [AlphaBlend](#) [SetTextOverlay](#) [AlphaBlendValue](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)

[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Full-featured EBook editor

## SetTextOverlay\_Scrolling

### TVideoGrabber.SetTextOverlay\_Scrolling

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the text scrolling for the specified text overlay

#### Declaration

procedure SetTextOverlay\_Scrolling (Index: LongInt; Value: Boolean);

void \_\_fastcall SetTextOverlay\_Scrolling(int Index, bool Value);

#### Description

Enable the scrolling of the text overlay.

The scrolling speed is adjusted by [SetTextOverlay\\_ScrollingSpeed](#).

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay\\_StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Create High-Quality Documentation with a Help Authoring Tool

## SetTextOverlay\_ScrollingSpeed

### TVideoGrabber.SetTextOverlay\_ScrollingSpeed

[Prev](#)[Next](#)[TVideoGrabber](#) [Methods](#)

Activates the text scrolling for the specified text overlay

#### Declaration

procedure SetTextOverlay\_ScrollingSpeed (Index: LongInt; Value: LongInt);

void \_\_fastcall SetTextOverlay\_ScrollingSpeed(int Index, int Value);

#### Description

Used to adjust the scrolling speed of the text overlay.

The scrolling must be enabled first with [SetTextOverlay\\_Scrolling](#).

Value = 0 -> no scrolling

Value > 0 -> scrolling from right to left

Value < 0 -> scrolling from left to right

E.g.:

SetTextOverlay\_ScrollingSpeed (index, 1) -> slow scrolling

SetTextOverlay\_ScrollingSpeed (index, -3) -> backwards average scrolling

SetTextOverlay\_ScrollingSpeed (index, 6) -> fast scrolling

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEODHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)  
[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

## SetTextOverlay\_Shadow

### TVideoGrabber.SetTextOverlay\_Shadow

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables the shadow for the specified text overlay

#### Declaration

procedure SetTextOverlay\_Shadow (Index: LongInt; Value: Boolean);

void \_\_fastcall SetTextOverlay\_Shadow(int Index, bool Value);

#### Description

Enables the shadow under the text overlay.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Free PDF documentation generator

## SetTextOverlay\_ShadowColor

### TVideoGrabber.SetTextOverlay\_ShadowColor

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the color of the shadow for the specified text overlay

#### Declaration

procedure SetTextOverlay\_ShadowColor (Index: LongInt; Value: TColor);

void \_\_fastcall SetTextOverlay\_ShadowColor(int Index, Graphics::TColor Value);

#### Description



Specifies the color of the shadow under the text overlay when the shadow has been enabled with [SetTextOverlay\\_Shadow](#).

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Transform Your Help Documentation Process with a Help Authoring Tool

## SetTextOverlay\_ShadowDirection

### TVideoGrabber.SetTextOverlay\_ShadowDirection

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the direction of the shadow for the specified text overlay

#### Declaration

```
procedure SetTextOverlay_ShadowDirection(Index: LongInt; Value: TCardinalDirection);
```

```
void __fastcall SetTextOverlay_ShadowDirection(int Index, TCardinalDirection Value);
```

#### Description

Specifies the direction of the shadow under the text overlay when the shadow has been enabled with [SetTextOverlay\\_Shadow](#).

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)

[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#)  
[SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#)  
[SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#)  
[SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#)  
[SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#)  
[SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#)  
[TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#)  
[TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#)  
[TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#)  
[TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#)  
[TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc

## SetTextOverlay\_String

### TVideoGrabber.SetTextOverlay\_String

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets or retrieves the text string that will be drawn over video frames for the specified text overlay

#### Declaration

procedure SetTextOverlay\_String (Index: LongInt; Value: string);

void \_\_fastcall SetTextOverlay\_String(int Index, wchar\_t \*Value);

#### Description

Used to set or retrieve the text string that will be drawn over video frames.

The text string can be composed of several lines.

This property can be modified at any time, even if the text is currently drawn over a video frame.

Several TVideoGrabber variables can be used within the text string, delimited by a percent symbol **(2)**. When TVideoGrabber detects one of these variables the variable label is replaced by its current value:

"%sys\_time[dd/mm/yy hh:nn:ss]%" : current system date/time **(1)**

"%dv\_time[dd/mm/yy hh:nn:ss]%" : current date/time stored on the DV VCR tape **(1)**

"%time\_code%" : current DV VCR time code, if available

"%frame\_count%" : number of the current frame

"%time\_full%" : time of the current frame in hh:mm:ss:cc format

"%time\_sec%" : time of the current frame, in seconds with 2 decimals

"%time\_100ns%" : time of the current frame, in 100 nano-seconds units

"%custom0% to %custom9%" : up to 10 custom variables that can be set by using

[SetTextOverlay\\_CustomVar](#) .

(1) any valid [date/time format](#) is accepted between the brackets of sys\_time and dv\_time.

(2) the percent symbol is a reserved character. If you need to display the percent symbol itself, just duplicate it, e.g. TextOverlay\_String = "the percent symbol is %%"

See [SetTextOverlay\\_CustomVar](#)

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#)  
[GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)

[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [AlphaBlend](#) [SetTextOverlay](#) [AlphaBlendValue](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TextOverlay](#) [VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Free EBook and documentation generator

## SetTextOverlay\_TargetDisplay

### TVideoGrabber.SetTextOverlay\_TargetDisplay

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to specify the location (video frame or window) of the text overlay

#### Declaration

procedure SetTextOverlay\_TargetDisplay (Index: LongInt; Value: LongInt);

void \_\_fastcall SetTextOverlay\_TargetDisplay(int Index, int Value);

#### Description

Used to specify the location (video frame or window) of the text overlay specified by its index

See [Frame overlay vs window overlay](#)

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay](#) [AlphaBlend](#)  
[GetTextOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [AlphaBlend](#) [SetTextOverlay](#) [AlphaBlendValue](#) [SetTextOverlay](#) [BkColor](#)



[SetTextOverlay\\_CustomVar](#)
[SetTextOverlay\\_Enabled](#)
[SetTextOverlay\\_Font](#)
[SetTextOverlay\\_FontColor](#)
[SetTextOverlay\\_GradientColor](#)
[SetTextOverlay\\_GradientMode](#)
[SetTextOverlay\\_HighResFont](#)
[SetTextOverlay\\_Left](#)
[SetTextOverlay\\_Right](#)
[SetTextOverlay\\_Scrolling](#)
[SetTextOverlay\\_ScrollingSpeed](#)
[SetTextOverlay\\_Shadow](#)
[SetTextOverlay\\_ShadowColor](#)
[SetTextOverlay\\_ShadowDirection](#)
[SetTextOverlay\\_String](#)
[SetTextOverlay\\_Top](#)
[SetTextOverlay\\_Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay\\_Align](#)
[TextOverlay\\_BkColor](#)
[TextOverlay\\_CreateCustomFont](#)
[TextOverlay\\_CreateCustomFont2](#)
[TextOverlay\\_Enabled](#)
[TextOverlay\\_Font](#)
[TextOverlay\\_FontColor](#)
[TextOverlay\\_Left](#)
[TextOverlay\\_Right](#)
[TextOverlay\\_Scrolling](#)
[TextOverlay\\_ScrollingSpeed](#)
[TextOverlay\\_Selector](#)
[TextOverlay\\_Shadow](#)
[TextOverlay\\_ShadowColor](#)
[TextOverlay\\_ShadowDirection](#)
[TextOverlay\\_String](#)
[TextOverlay\\_Top](#)
[TextOverlay\\_Transparent](#)
[TextOverlay\\_VideoAlignment](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface](#)

## SetTextOverlay\_Top

### TVideoGrabber.SetTextOverlay\_Top

[Prev](#)
[Next](#)

[TVideoGrabber](#)
[Methods](#)

Sets the top position (in pixels) where the text will be drawn over video frames for the specified text overlay

#### Declaration

procedure SetTextOverlay\_Top (Index: LongInt; Value: LongInt);

void \_\_fastcall SetTextOverlay\_Top(int Index, int Value);

#### Description

Used to set or retrieve the top position (in pixels) where the text will be drawn over video frames.

#### See Also

[TCardinalDirection](#)
[TOnFrameOverlayUsingDC](#)
[TOnFrameOverlayUsingDIB](#)
[TTextOverlayAlign](#)
[AdjustOverlayAspectRatio](#)
[DrawBitmapOverFrame](#)
[GetFrameInfoString](#)
[GetTextOverlay\\_AlphaBlend](#)
[GetTextOverlay\\_AlphaBlendValue](#)
[ImageOverlay](#)
[StretchToVideoSize](#)
[MouseWheelEventEnabled](#)
[OnFrameBitmap](#)
[OnFrameBitmapEventSynchronise](#)
[OnFrameOverlayUsingDC](#)
[OnFrameOverlayUsingDIB](#)
[OnFrameOverlayUsingVIDEONHDR](#)
[OnMouseDown](#)
[OnMouseMove](#)
[OnMouseUp](#)
[OnMouseWheel](#)
[OverlayAfterTransform](#)
[RefreshPlayerOverlays](#)
[SetImageOverlay\\_AlphaBlend](#)
[SetImageOverlay\\_AlphaBlendValue](#)
[SetImageOverlay\\_ChromaKey](#)
[SetImageOverlay\\_ChromaKeyLeewayPercent](#)
[SetImageOverlay\\_ChromaKeyRGBColor](#)
[SetImageOverlay\\_Enabled](#)
[SetImageOverlay\\_Height](#)
[SetImageOverlay\\_LeftLocation](#)
[SetImageOverlay\\_RotationAngle](#)
[SetImageOverlay\\_StretchToVideoSize](#)
[SetImageOverlay\\_TargetDisplay](#)
[SetImageOverlay\\_TopLocation](#)
[SetImageOverlay\\_Transparent](#)
[SetImageOverlay\\_TransparentColorValue](#)
[SetImageOverlay\\_UseTransparentColor](#)
[SetImageOverlay\\_Width](#)
[SetImageOverlayFromBMPFile](#)
[SetImageOverlayFromHBitmap](#)
[SetImageOverlayFromHBitmap2](#)
[SetImageOverlayFromImageFile](#)
[SetImageOverlayFromImageFile2](#)
[SetImageOverlayFromJPEGFile](#)
[SetImageOverlayFromTBitmap](#)
[SetImageOverlayFromTBitmap2](#)
[SetImageOverlayFromTImage](#)
[SetImageOverlayFromTImage2](#)
[SetTextOverlay\\_Align](#)
[SetTextOverlay\\_AlphaBlend](#)
[SetTextOverlay\\_AlphaBlendValue](#)
[SetTextOverlay\\_BkColor](#)
[SetTextOverlay\\_CustomVar](#)
[SetTextOverlay\\_Enabled](#)
[SetTextOverlay\\_Font](#)
[SetTextOverlay\\_FontColor](#)
[SetTextOverlay\\_GradientColor](#)
[SetTextOverlay\\_GradientMode](#)
[SetTextOverlay\\_HighResFont](#)
[SetTextOverlay\\_Left](#)
[SetTextOverlay\\_Right](#)
[SetTextOverlay\\_Scrolling](#)
[SetTextOverlay\\_ScrollingSpeed](#)
[SetTextOverlay\\_Shadow](#)
[SetTextOverlay\\_ShadowColor](#)
[SetTextOverlay\\_ShadowDirection](#)
[SetTextOverlay\\_String](#)
[SetTextOverlay\\_TargetDisplay](#)
[SetTextOverlay\\_Transparent](#)
[ShapeOverlay](#)
[ShapeOverlayEnabled](#)
[ShapeOverlayList](#)
[TextOverlay\\_Align](#)
[TextOverlay\\_BkColor](#)
[TextOverlay\\_CreateCustomFont](#)
[TextOverlay\\_CreateCustomFont2](#)
[TextOverlay\\_Enabled](#)
[TextOverlay\\_Font](#)
[TextOverlay\\_FontColor](#)
[TextOverlay\\_Left](#)
[TextOverlay\\_Right](#)
[TextOverlay\\_Scrolling](#)
[TextOverlay\\_ScrollingSpeed](#)
[TextOverlay\\_Selector](#)
[TextOverlay\\_Shadow](#)
[TextOverlay\\_ShadowColor](#)
[TextOverlay\\_ShadowDirection](#)
[TextOverlay\\_String](#)
[TextOverlay\\_Top](#)
[TextOverlay\\_Transparent](#)
[TextOverlay\\_VideoAlignment](#)
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Why Microsoft Word Isn't Cut Out for Documentation: The Benefits of a Help Authoring Tool](#)

## SetTextOverlay\_Transparent

### TVideoGrabber.SetTextOverlay\_Transparent

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables/disables transparency of text drawn over video frames for the specified text overlay

#### Declaration

procedure SetTextOverlay\_Transparent (Index: LongInt; Value: Boolean);

```
void __fastcall SetTextOverlay_Transparent(int Index, bool Value);
```

#### Description

Used to enable or disable transparency of text drawn over video frames.  
If enabled, any value set with [SetTextOverlay\\_BkColor](#) is ignored.

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Make Your PDFs More Secure with Encryption and Password Protection

## SetVideoCompressionDefaults

### TVideoGrabber.SetVideoCompressionDefaults

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Resets the current [video compressor](#) .

#### Declaration

function SetVideoCompressionDefaults: Boolean;

**bool** \_\_fastcall SetVideoCompressionDefaults(**void**)

Function SetVideoCompressionDefaults() As Boolean

### Description

Used to reset the compression settings of the current [video compressor](#) to their default values.

### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## SetVideoCompressionSettings

### TVideoGrabber.SetVideoCompressionSettings

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets video compression settings.

### Declaration

**function** SetVideoCompressionSettings(DataRate, KeyFrameRate, PFramesPerKeyFrame, WindowSize: LongInt; Quality: Double): Boolean;

**bool** \_\_fastcall SetVideoCompressionSettings(**int** DataRate, **int** KeyFrameRate, **int** PFramesPerKeyFrame, **int** WindowSize, **double** Quality)

Function SetVideoCompressionSettings(DataRate As Long, KeyFrameRate As Long, PFramesPerKeyFrame As Long, WindowSize As Long, Quality As Double) As Boolean

### Description

Sets the video compression settings supported by some codecs. Returns true if the current video compressor supports this interface.

The current values can be retrieved by [GetVideoCompressionSettings](#) .

**DataRate:** retrieves the output data rate.

**KeyFrameRate:** The key-frame rate is the number of frames per key frame. For example, if the rate is 15, then a key frame occurs every 15 frames.

**PFramesPerKeyFrame:** P frames are used only in MPEG compression. E.g. let's say a key frame occurs once every 10 frames, and there are three P frames per key frame. The P frames will be spaced evenly between the key frames. The remaining six frames are bi-directional (B) frames.

**WindowSize:** retrieves the number of frames over which the compressor will maintain the average data rate. E.g. if a data rate of 100K/sec and a frame rate of 10 frames per second, if the window size is 1, then every frame will be 10K or less. If the window size is 5, then every five consecutive frames will average 10K per frame, but individual frames may exceed this size.

**Quality:** The quality is expressed as a value between 0.0 and 1.0, where 1.0 indicates the best quality and 0.0 indicates the worst quality. If the value is negative, the filter will use the default quality.

**CanCrunch:** the compressor can compress video to a specified data rate (see DataRate above).

**CanKeyFrame:** the compressor supports the KeyFrame property above.

**CanBFrame:** the compressor supports the PFramesPerKeyFrame property above.

**CanWindow:** the compressor supports WindowSize property above.

**CanQuality:** the compressor supports Quality property above.

### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#)

[AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

## SetVideoControlMode

### TVideoGrabber.SetVideoControlMode

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the [TVideoControl](#) settings

#### Declaration

**function** SetVideoControlMode(FlipHorizontal, FlipVertical, ExternalTriggerEnable, Trigger: Boolean): Boolean;

**bool** \_\_fastcall SetVideoControlMode(**bool** FlipHorizontal, **bool** FlipVertical, **bool** ExternalTriggerEnable, **bool** Trigger)

Function SetVideoControlMode (FlipHorizontal As Boolean, FlipVertical As Boolean, ExternalTriggerEnable As Boolean, Trigger As Boolean)As Boolean

#### Description

Used to set all the [TVideoControl](#) settings, if available for the current video capture device (see [IsVideoControlModeAvailable](#) ).

#### See Also

[TVideoControl](#) [GetVideoControlMode](#) [IsVideoControlAvailable](#) [IsVideoControlModeAvailable](#) [SetVideoControlMode2](#) [VideoControlSettings](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## SetVideoControlMode2

### TVideoGrabber.SetVideoControlMode2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets one of the [TVideoControl](#) settings

#### Declaration

**function** SetVideoControlMode(Mode: TVideoControl; Value: Boolean): Boolean;

**bool** \_\_fastcall SetVideoControlMode2(TVideoControl Mode, **bool** Value);

Function SetVideoControlMode (mode as TVideoControl, Value as Boolean) As Boolean

#### Description

Used to set one of the [TVideoControl](#) settings, if available for the current video capture device (see [IsVideoControlModeAvailable](#) ).

#### See Also

[TVideoControl](#) [GetVideoControlMode](#) [IsVideoControlAvailable](#) [IsVideoControlModeAvailable](#) [SetVideoControlMode](#) [VideoControlSettings](#)

## SetVideoQuality

### TVideoGrabber.SetVideoQuality

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a [TVideoQuality](#) value.

#### Declaration

**function** SetVideoQuality(Setting: TVideoQuality; SetAuto: Boolean; SetDefault: Boolean; SetValue: LongInt): Boolean;

**bool** \_\_fastcall SetVideoQuality(TVideoQuality Setting, **bool** SetAuto, **bool** SetDefault, **int** SetValue)

Function SetVideoQuality(Setting As TVideoQuality, SetAuto as Boolean, SetDefault as Boolean, SetValue As Long) As Boolean

#### Description

Used to set a [TVideoQuality](#) value, if available for the current video capture device (test [IsVideoQualitySettingAvailable](#) for availability).

- if **SetAuto** is true, SetDefault and SetValue are ignored and the setting is switched in "auto" mode
- if **SetAuto** is false and **SetDefault** is true, SetValue is ignored and the setting is set to its default value,
- if **SetAuto** is false and **SetDefault** is false, the **SetValue** value is applied to the setting.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [VideoQualityAuto](#) [VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

## SetVMR9ImageAdjustmentValue

### TVideoGrabber.SetVMR9ImageAdjustmentValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a VMR9 image adjustment setting.

#### Declaration

**function** SetVMR9ImageAdjustmentValue(MainDisplay: Boolean; VMR9ControlSetting: TVMR9ImageAdjustment; Value: LongInt; FixRange: Boolean): Boolean;

**bool** \_\_fastcall SetVMR9ImageAdjustmentValue(**bool** MainDisplay, TVMR9ImageAdjustment VMR9ControlSetting, **int** Value, **bool** FixRange)

Function SetVMR9ImageAdjustmentValue(MainDisplay As Boolean, VMR9ControlSetting As TVMR9ImageAdjustment, param3 As Long, FixRange As Boolean) As Boolean

#### Description

Used to set a VMR9 image adjustment property (brightness, contrast, hue, saturation). The VMR9ControlSetting parameter is a [TVMR9ImageAdjustment](#) type.

See also [GetVMR9ImageAdjustmentValue](#).

**See Also**

[GetVMR9ImageAdjustmentBounds](#) [IsVMR9ImageAdjustmentAvailable](#) [TVMR9ImageAdjustment](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize your documentation process with HelpNDoc's online capabilities](#)

## SetVuMeter\_Enabled

### TVideoGrabber.SetVuMeter\_Enabled

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to enable/disable a VU-meter

**Declaration**

procedure TVideoGrabber.SetVuMeter\_Enabled (Index: LongInt; Value: TVuMeter);

void \_\_fastcall SetVuMeter\_Enabled(int Index, TVuMeter Value);

**Description**

Enables/disables the VU-meter specified by its index

**See Also**

[SetVUMeterSetting](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## SetVUMeterSetting

### TVideoGrabber.SetVUMeterSetting

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to adjust the VU-Meter settings.

**Declaration**

function SetVUMeterSetting (ChannelIndex: LongWord; VUMeterSetting: TVUMeterSetting; Value: LongWord): Boolean;

bool \_\_fastcall SetVUMeterSetting (TVUMeterSetting VUMeterSetting, **unsigned int** Value)

Function SetVUMeterSetting (Setting As TxVUMeterSetting, Value as Long) As Boolean

**Description**

Used to adjust the VU-Meter settings, of the [TVUMeterSetting](#) type.

**vu\_Handle** : sets the Handle of the panel or image control on which the VUMeter will be displayed (0 disables the previous handle).

**vu\_WarningPercent** : percentage of the level above which the vu\_WarningColor is used

**vu\_PeakPercent** : percentage of the level above which the vu\_PeakColor is used

**vu\_BkgndColor** : color of the VU-meter background

**vu\_NormalColor** : color of the normal level (green by default)

**vu\_WarningColor** : color of the warning level (above the vu\_WarningPercent)

**vu\_PeakColor** : color of the peak level (above the vu\_PeakPercent)

**vu\_TickSize** : size of ticks in the bargraph VU-Meter

**vu\_TickInterval** : interval between ticks in the bargraph VU-Meter

**vu\_NeedleThickness** : thickness of the needle in the analog VU-Meter

See the "[Audio levels and VU-Meters](#)" chapter that explains how to use the VU-Meters.



## See Also

[SetVuMeter.Enabled](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

## SetWindowRecordingByHandle

### TVideoGrabber.SetWindowRecordingByHandle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to specify a window for screen recording

#### Declaration

**function** SetWindowRecordingByHandle (WindowHandle: LongInt): **Boolean**;

**bool** \_\_fastcall SetWindowRecordingByHandle(**int** WindowHandle);

Function SetWindowRecordingByHandle (WindowHandle as Long) as Boolean

#### Description

Used for the screen recording to specify a window to record by its handle.  
The handle of this window can be retrieved by invoking [EnumerateWindows](#).

See the "[Recording a window](#)" chapter for more information.

## See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## SetWindowRecordingByName

### TVideoGrabber.SetWindowRecordingByName

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Used to specify a window for screen recording

#### Declaration

**function** SetWindowRecordingByName (WindowName: string; ExactMatch: Boolean): **Boolean**;

**bool** \_\_fastcall SetWindowRecordingByName(wchar\_t \*WindowName, **bool** ExactMatch);

Function SetWindowRecordingByName (WindowName as String, ExactMatch as Boolean) as Boolean

#### Description

Used for the screen recording to specify a window to record by its window name or class name.

The window name or class name of this window can be picked up by looking at the title bar of the window, or by retrieved by invoking [EnumerateWindows](#).

- you can identify the exact window name by passing the full string to the WindowName parameter and setting ExactMatch = true
- you can identify the window by a substring of its name, passed to the WindowName parameter, and by setting ExactMatch = false.



Note: when `ExactMatch = false`, the `WindowName` parameter is not case-sensitive.

See the "[Recording a window](#)" chapter for more information.

#### See Also

[TOnEnumerateWindows](#) [EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)

## SetWindowTransparency

### TVideoGrabber.SetWindowTransparency

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets the transparency (color key or alpha value) on a window or form

#### Declaration

**function** SetWindowTransparency (WndHandle: HWND; UseColorKey: Boolean; ColorKeyValue: TColor; UseAlpha: Boolean; AlphaValue: LongInt): Boolean;

**bool** **\_\_fastcall** SetWindowTransparency (HWND WndHandle, **bool** UseColorKey, COLORREF ColorKeyValue, **bool** UseAlpha, byte AlphaValue)

function SetWindowTransparency (WndHandle as Long, UseColorKey as Boolean, ColorKeyValue as Long, UseAlpha as Boolean, AlphaValue as Long) as Boolean

#### Description

Used to set the transparency (color key or alpha value) on a window or form.

Set [ColorKeyEnabled](#) = true, then invoke this function and pass the handle to the form you wish to be transparent.

*You can find sample code in the transparency checkbox code of the "display" tab of the MainDemo project.*

E.g. in Delphi:

```
procedure TfrmMainForm.chkTransparencyClick(Sender: TObject);
begin
    // in this sample the transparent window is created in the VideoGrabberColorKeyChange
    // event below when the color key is set
    VideoGrabber.ColorKeyEnabled := chkTransparency.Checked;
    if VideoGrabber.ColorKeyEnabled then begin
        if not assigned (FTransparentForm) then begin
            Application.CreateForm (TForm, FTransparentForm);
            FTransparentForm.Width := 300;
            FTransparentForm.Height := 100;
            FTransparentForm.Position := poScreenCenter;
            FTransparentForm.FormStyle := fsStayOnTop;
            FTransparentPanel := TPanel.Create (FTransparentForm);
            FTransparentPanel.Parent := FTransparentForm;
            FTransparentPanel.BevelInner := bvNone;
            FTransparentPanel.BevelOuter := bvNone;
            FTransparentPanel.Align := alClient;
            FTransparentPanel.Caption := 'transparent window';
            FTransparentPanel.Font.Color := clRed;
            FTransparentPanel.Font.Size := 16;
```

```

end;

FTransparentForm.Color := clBlue;
FTransparentPanel.Color := VideoGrabber.ColorKey;
VideoGrabber.SetWindowTransparency (FTransparentForm.Handle, true, VideoGrabber.Col
FTransparentForm.Show;
end;
end;

```

### See Also

[Dual display TVideoRenderer AdjustPixelAspectRatio Display Active Display AlphaBlendEnabled Display AlphaBlendValue Display AutoSize Display Embedded Display FullScreen Display Height Display Left Display Monitor Display MouseMovesWindow Display PanScanRatio Display SetLocation Display StayOnTop Display Top Display TransparentColorEnabled Display TransparentColorValue Display VideoHeight Display VideoPortEnabled Display VideoWidth Display VideoWindowHandle Display Width DualDisplay Active DualDisplay AlphaBlendEnabled DualDisplay AlphaBlendValue DualDisplay AutoSize DualDisplay Embedded DualDisplay FullScreen DualDisplay Height DualDisplay Left DualDisplay Monitor DualDisplay MouseMovesWindow DualDisplay PanScanRatio DualDisplay SetLocation DualDisplay StayOnTop DualDisplay Top DualDisplay TransparentColorEnabled DualDisplay TransparentColorValue DualDisplay VideoHeight DualDisplay VideoPortEnabled DualDisplay VideoWidth DualDisplay VideoWindowHandle DualDisplay Visible DualDisplay Width IsVideoPortAvailable Monitor Primary Index MonitorBounds MonitorsCount OnLeavingFullScreen SetParentWindow VideoDoubleBuffered VideoHeight PreferredAspectRatio VideoRendererExternal VideoRendererExternalIndex VideoRendererPriority VideoVisibleWhenStopped VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## ShapeOverlayList

### TVideoGrabber.ShapeOverlayList

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

List of TShape components that have to be drawn over video frames.

#### Declaration

**function** ShapeOverlayList(Shape: TShape; Add: Boolean): Boolean;

**bool** \_\_fastcall ShapeOverlayList(ExtCtrls::TShape \*Shape, **bool** Add)

n/a

#### Description

Used to add/remove TShape components in the list of TShape components that will be drawn over video frames when [ShapeOverlayEnabled](#) is true.

- to add a TShape, put a TShape component (or create it dynamically at runtime), and then simply call [ShapeOverlayList](#) (YourShapeComponent, true).
- to remove a TShape component from the list, call [ShapeOverlayList](#) (YourShapeComponent, false).
- the function returns true upon success.

Each The TShape component in the list will be drawn at its Left, Top, Right, Height coordinates if its Enabled property is true.

It is possible to disable its Visible property whether it must not be visible on the form.

Note: to use only a single TShape component, it is possible to assign it to the [ShapeOverlay](#) property, instead of using this function.

**See Also**

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#)  
[OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlay](#) [FromBMPFile](#)  
[SetImageOverlay](#) [FromHBitmap](#) [SetImageOverlay](#) [FromHBitmap2](#) [SetImageOverlay](#) [FromImageFile](#)  
[SetImageOverlay](#) [FromImageFile2](#) [SetImageOverlay](#) [FromJPEGFile](#) [SetImageOverlay](#) [FromTBitmap](#)  
[SetImageOverlay](#) [FromTBitmap2](#) [SetImageOverlay](#) [FromTImage](#) [SetImageOverlay](#) [FromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#)  
[TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#)  
[TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#)  
[TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#)  
[TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

**ShowDebugWindow****TVideoGrabber.ShowDebugWindow**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Opens a window showing the TVideoGrabber log

**Declaration**

**procedure** ShowDebugWindow;

**void** \_\_fastcall ShowDebugWindow(**void**)

Sub ShowDebugWindow()

**Description**

Invoke ShowDebugWindow() for debugging purpose.

This will popup a window that will report the TVideoGrabber log events.

This may be helpful for tracking more easily the errors reported by TVideoGrabber.

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

**ShowDialog****TVideoGrabber.ShowDialog**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Display a [TDialog](#) dialog.

#### Declaration

**function** ShowDialog(Dialog: TDialog): Boolean;

**bool** \_\_fastcall ShowDialog(TDialog Dialog)

Function ShowDialog(Dialog As TxDialog) As Boolean

#### Description

Used to display a [TDialog](#) dialog, if available for the current video capture device (test [IsDialogAvailable](#) for availability).

#### See Also

[Player features](#) [TNetworkStreamingType](#) [TOnClientConnection](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [WDM drivers](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [TOnPlayerBufferingData](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoConnectRelatedPins](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FastForwardPlayer](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [Last Clip](#) [Played](#) [MP4NeedsReindexing](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [OnNoVideoDevices](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OnVideoDeviceSelected](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [SourceStream](#) [StartAudioRendering](#) [StopPlayer](#) [StreamingURL](#) [SynchronizationRole](#) [Synchronized](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## StartAudioRecording

### TVideoGrabber.StartAudioRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts the recording of the audio stream only.

#### Declaration

**function** StartAudioRecording: Boolean;

**bool** \_\_fastcall StartAudioRecording(**void**)

Function StartAudioRecording() As Boolean

#### Description

Used to start the recording of the audio stream only by using the current [audio capture device](#).

However if the current [video capture device](#) has an audio output, it will be firstly used, so if you wish to be sure use the current [audio capture device](#) and not the possible audio stream coming out the current [video](#)

[capture device](#), set [VideoDevice](#) = -1 to keep the video devices disabled.

When the **HoldRecording** parameter is set to "false", the recording starts as soon as the recording graph is built (after a variable number of seconds, depending of the video capture device, audio/video compressors, etc...).

When the **HoldRecording** parameter is set to "true", the recording graph is built and started in preview mode, and the recording is held until [ResumeRecording](#) has been invoked.

When the recording graph is built, the [OnRecordingReadyToStart](#) event occurs to let you know that you can invoke [ResumeRecording](#) when you want.

See the [recording](#) chapter for more information.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation a Breeze with a Help Authoring Tool](#)

## StartAudioRendering

### TVideoGrabber.StartAudioRendering

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts the rendering of the audio stream only.

#### Declaration

**function** StartAudioRendering: Boolean;

**bool** \_\_fastcall StartAudioRendering(**void**)

Function StartAudioRendering() As Boolean

#### Description

Used to start the rendering of the audio stream only by using the current [audio capture device](#).

However if the current [video capture device](#) has an audio output, it will be firstly used, so if you wish to be sure use the current [audio capture device](#) and not the possible audio stream coming out the current [video capture device](#), set [VideoDevice](#) = -1 to keep the video devices disabled.

This function designed to activate the audio broadcasting to the network. See the [Audio broadcasting](#) chapter.

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)

[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Make the switch to CHM with HelpNDoc's hassle-free WinHelp HLP to CHM conversion tool](#)

## StartPreview

### TVideoGrabber.StartPreview

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts preview.

#### Declaration

**function** StartPreview: Boolean;

**bool** \_\_fastcall StartPreview(**void**)

Function StartPreview() As Boolean

#### Description

Used to start the preview using the current [video capture device](#) (and [audio capture device](#) , whether [AudioRendering](#) is enabled).

See [Preview overview](#) .

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## StartRecording

### TVideoGrabber.StartRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts the recording.

#### Declaration

**function** StartRecording: Boolean;

**bool** \_\_fastcall StartRecording(**void**)

Function StartRecording() As Boolean

#### Description

Used to start the recording by using the current [video capture device](#) (and [audio capture device](#) , if [AudioRecording](#) is enabled).

When the **HoldRecording** parameter is set to "false", the recording starts as soon as the recording graph is built (after a variable number of seconds, depending of the video capture device, audio/video compressors, etc...).



When the **HoldRecording** parameter is set to "true", the recording graph is built and started in preview mode, and the recording is held until [ResumeRecording](#) has been invoked.

When the recording graph is built, the [OnRecordingReadyToStart](#) event occurs to let you know that you can invoke [ResumeRecording](#) when you want.

See the [recording](#) chapter for more information.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Make Documentation a Breeze with HelpNDoc's Clean and Efficient User Interface

## StartReencoding

### TVideoGrabber.StartReencoding

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts the reencoding of a video clip.

#### Declaration

**function** StartReencoding: Boolean;

**bool** \_\_fastcall StartReencoding(**void**)

Function StartReencoding() As Boolean

#### Description

Used to starts the reencoding of a video clip, according to the Reencoding\_... properties. See the [Reencoding video clips](#) chapter for more information.

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding](#) [IncludeAudioStream](#) [Reencoding](#) [IncludeVideoStream](#) [Reencoding](#) [Method](#) [Reencoding](#) [NewVideoClip](#) [Reencoding](#) [SourceVideoClip](#) [Reencoding](#) [StartFrame](#) [Reencoding](#) [StartTime](#) [Reencoding](#) [StopFrame](#) [Reencoding](#) [StopTime](#) [Reencoding](#) [UseAudioCompressor](#) [Reencoding](#) [UseFrameGrabber](#) [Reencoding](#) [UseVideoCompressor](#) [Reencoding](#) [WMVOutput](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer

## StartSynchronized

### TVideoGrabber.StartSynchronized

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)



Starts the preview or recording simultaneously on several components

#### Declaration

**function** StartSynchronized: Boolean;

**bool** \_\_fastcall StartSynchronized(**void**)

Function StartSynchronized() As Boolean

#### Description

Used to start the preview or the recording at the same time on several TVideoGrabber components.

See the [Synchronization of several TVideoGrabber components](#) chapter.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnFrameProgress](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TOnResizeVideo](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AutoRefreshPreview](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsRecordingPaused](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnResizeVideo](#) [PausePreview](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [PreviewZoomSize](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumePreview](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartPreview](#) [StartRecording](#) [StopPreview](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoPlayableWhileRecording](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

---

Created with the Standard Edition of HelpNDoc: [Ensure High-Quality Documentation with HelpNDoc's Hyperlink and Library Item Reports](#)

---

## Stop

### TVideoGrabber.Stop

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stops any task currently running.

#### Declaration

**function** Stop: Boolean;

**void** \_\_fastcall Stop(**void**)

Function Stop() As Boolean

#### Description

Stops any task currently running (preview, recording, playback as well as reencoding).  
This function can be used instead of [StopPreview](#), [StopRecording](#), [StopReencoding](#) or [ClosePlayer](#).  
Returns true upon success.

---

Created with the Standard Edition of HelpNDoc: [Create Professional CHM Help Files with HelpNDoc's](#)

---

## StopPlayer

### TVideoGrabber.StopPlayer

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stops playing a video clip.

#### Declaration

**procedure** StopPlayer;

**void** \_\_fastcall StopPlayer(**void**)

Sub StopPlayer()

#### Description

Used to stop a video clip currently playing.  
See the [Media Player](#) chapter.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [HelpNDoc's Project Analyzer: Incredible documentation assistant](#)

## StopPreview

### TVideoGrabber.StopPreview

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stops preview.

#### Declaration

**procedure** StopPreview;

**void** \_\_fastcall StopPreview(**void**)

Sub StopPreview()

#### Description

Used to stop the current preview, previously started by [StartPreview](#) .

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#)

[PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## StopRecording

### TVideoGrabber.StopRecording

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stops the current recording.

#### Declaration

**procedure** StopRecording;

**void** \_\_fastcall StopRecording(**void**)

Sub StopRecording()

#### Description

Used to stop the current recording, previously started by [StartRecording](#) .

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Eliminate the Struggles of Documentation with a Help Authoring Tool](#)

## StopReencoding

### TVideoGrabber.StopReencoding

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stops the reencoding of a video clip.

#### Declaration

**function** StopReencoding: Boolean;

**bool** \_\_fastcall StopReencoding(**void**)

Function StopReencoding() As Boolean

#### Description

Used to stop the reencoding of a video clip before the process ends.  
See the [Reencoding video clips](#) chapter for more information.

## See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding\\_IncludeAudioStream](#) [Reencoding\\_IncludeVideoStream](#) [Reencoding\\_Method](#) [Reencoding\\_NewVideoClip](#) [Reencoding\\_SourceVideoClip](#) [Reencoding\\_StartFrame](#) [Reencoding\\_StartTime](#) [Reencoding\\_StopFrame](#) [Reencoding\\_StopTime](#) [Reencoding\\_UseAudioCompressor](#) [Reencoding\\_UseFrameGrabber](#) [Reencoding\\_UseVideoCompressor](#) [Reencoding\\_WMVOutput](#) [StartReencoding](#)

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

## StreamInterface\_PushData

### TVideoGrabber.StreamInterface\_PushData

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Pushes the raw samples into the stream.

#### Declaration

**function** StreamInterface\_PushData (Buffer: pByte; BufferSize: LongInt): LongInt;

**int** \_\_fastcall StreamInterface\_PushData(System::PByte Buffer, **int** BufferSize);

#### Description

Invoke this function to push the raw samples (e.g. H264) to (re)build the stream.  
Look at the [Stream Interface](#) chapter for more information.

## See Also

[Stream Interface](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

## TextOverlay\_CreateCustomFont

### TVideoGrabber.TextOverlay\_CreateCustomFont

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Lets create a custom font for the current text overlay selected with [TextOverlay\\_Selector](#)

#### Declaration

**function** TextOverlay\_CreateCustomFont(fHeight: LongInt; fWidth: LongInt; fEscapement: LongInt; fOrientation: LongInt; fWeight: LongInt; fItalic: Boolean; fUnderline: Boolean; fStrikeOut: Boolean; fCharSet: LongWord\_; fOutputPrecision: LongWord\_; fClipPrecision: LongWord\_; fQuality: LongWord\_; fPitchAndFamily: LongWord\_; FontFacename: **string**): Boolean;

**bool** \_\_fastcall TextOverlay\_CreateCustomFont(**int** fHeight, **int** fWidth, **int** fEscapement, **int** fOrientation, **int** fWeight, **bool** fItalic, **bool** fUnderline, **bool** fStrikeOut, **unsigned** fCharSet, **unsigned** fOutputPrecision, **unsigned** fClipPrecision, **unsigned** fQuality, **unsigned** fPitchAndFamily, System::wchar\_t \*FontFacename)

Function TextOverlay\_CreateCustomFont(fHeight as Long, fWidth as Long, fEscapement as Long, fOrientation as Long, fWeight as Long, fItalic as Boolean, fUnderline as Boolean, fStrikeOut as Boolean, fCharSet as Long, fOutputPrecision as Long, fClipPrecision as Long, fQuality as Long, fPitchAndFamily as Long, FontFacename as string) as Boolean

#### Description

Used to create a custom font used by the text overlay

E.g.:

```
VideoGrabber.TextOverlay_Selector = 4
VideoGrabber.TextOverlay_CreateCustomFont (16, 30, 0, 0, 200, true, false, false, 2, 0, 0, 0, 0,
"Wingdings")
VideoGrabber.TextOverlay_Enabled = true
```

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay\\_AlphaBlend](#) [GetTextOverlay\\_AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay\\_AlphaBlend](#) [SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#) [SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#) [SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#) [SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#) [SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#) [SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay\\_Align](#) [SetTextOverlay\\_AlphaBlend](#) [SetTextOverlay\\_AlphaBlendValue](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#) [SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#) [SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#) [SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#) [SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#) [SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TextOverlay\\_VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## TextOverlay\_CreateCustomFont2

### TVideoGrabber.TextOverlay\_CreateCustomFont2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Lets create a custom font for the text overlay specified by index

#### Declaration

**function** TextOverlay\_CreateCustomFont(Index: LongInt; fHeight: LongInt; fWidth: LongInt; fEscapement: LongInt; fOrientation: LongInt; fWeight: LongInt; fItalic: Boolean; fUnderline: Boolean; fStrikeOut: Boolean; fCharSet: LongWord\_; fOutputPrecision: LongWord\_; fClipPrecision: LongWord\_; fQuality: LongWord\_; fPitchAndFamily: LongWord\_; FontFacename: **string**): Boolean;

**bool** \_\_fastcall TextOverlay\_CreateCustomFont(int Index, int fHeight, int fWidth, int fEscapement, int fOrientation, int fWeight, **bool** fItalic, **bool** fUnderline, **bool** fStrikeOut, **unsigned** fCharSet, **unsigned** fOutputPrecision, **unsigned** fClipPrecision, **unsigned** fQuality, **unsigned** fPitchAndFamily, System::wchar\_t \*FontFacename)

Function TextOverlay\_CreateCustomFont(Index as Long, fHeight as Long, fWidth as Long, fEscapement as Long, fOrientation as Long, fWeight as Long, fItalic as Boolean, fUnderline as Boolean, fStrikeOut as Boolean, fCharSet as Long, fOutputPrecision as Long, fClipPrecision as Long, fQuality as Long, fPitchAndFamily as Long, FontFacename as string) as Boolean

#### Description

Used to create a custom font used by the text overlay having the specified index

E.g.: for the text overlay #5 (having the index 4):

```
VideoGrabber.TextOverlay_CreateCustomFont2 (4, 16, 30, 0, 0, 200, true, false, false, 2, 0, 0, 0, 0,
"Wingdings")
VideoGrabber.SetTextOverlay_Enabled (4, true)
```

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [GetTextOverlay](#) [AlphaBlend](#)  
[GetTextOverlay](#) [AlphaBlendValue](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [AlphaBlend](#) [SetTextOverlay](#) [AlphaBlendValue](#) [SetTextOverlay](#) [BkColor](#)  
[SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#)  
[SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#)  
[SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#)  
[SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#)  
[SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#)  
[ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#)  
[TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#)  
[TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#)  
[TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TextOverlay](#) [VideoAlignment](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily create HTML Help documents](#)

## ThirdPartyFilter\_AddToList

### TVideoGrabber.ThirdPartyFilter\_AddToList

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Adds a [third-party filter](#) to the list specified.

#### Declaration

**function** ThirdPartyFilter\_AddToList (Location: TThirdPartyFilterList; GUIDString: **string**;  
OptionalDLLFilePath: **string**; FilterName: **string**; Enable: Boolean; CanSaveFilterState: Boolean): Boolean;

**bool** \_\_fastcall ThirdPartyFilter\_AddToList(TThirdPartyFilterList Location, wchar\_t \*GUIDString, wchar\_t  
\*OptionalDLLFilePath, wchar\_t \*FilterName, **bool** Enable, **bool** CanSaveFilterState);

Function ThirdPartyFilter\_AddToList(Location As TxThirdPartyFilterList, GUIDString As String,  
OptionalDLLFilePath as String, FilterName As String, Enable As Boolean, SaveFilterState As Boolean) As  
Long

#### Description

Used to add a [third-party DirectShow filter](#) to a TThirdPartyFilterList list.

The function returns the index of the filter in the Location list. This index is necessary to access later the filter through other ThirdPartyFilter... functions.

Return -1 upon failure (e.g. if the filter is already in the list).

**Location:** [TThirdPartyFilterList](#) insertion point in the preview (or recording or play back) graph.



**GUID:** Guid of the filter to insert.

**Name:** name string associated to the filter (any value)

**Enable:** enables or disables the filter for the next preview (or recording or play back).

**SaveFilterState:** if true, the filter state will be saved in the registry and retrieved automatically for the next preview (or recording or play back).

#### See Also

[TThirdPartyFilterList](#) [TOnThirdPartyFilterConnected](#) [OnThirdPartyFilterConnected](#) [ThirdPartyFilter](#) [ClearList](#) [ThirdPartyFilter\\_Enable](#) [ThirdPartyFilter\\_RemoveFromList](#) [ThirdPartyFilter\\_ShowDialog](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## ThirdPartyFilter\_ClearList

### TVideoGrabber.ThirdPartyFilter\_ClearList

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Clears a third-party filter list.

#### Declaration

**function** ThirdPartyFilter\_ClearList(Location: TThirdPartyFilterList): Boolean;

**bool** \_\_fastcall ThirdPartyFilter\_ClearList(TThirdPartyFilterList Location)

Function ThirdPartyFilter\_ClearList(Location As TxThirdPartyFilterList) As Boolean

#### Description

Used to clear the third-party filter list specified.

**Location:** [TThirdPartyFilterList](#) location list to clear.

#### See Also

[TThirdPartyFilterList](#) [TOnThirdPartyFilterConnected](#) [OnThirdPartyFilterConnected](#) [ThirdPartyFilter\\_AddToList](#) [ThirdPartyFilter\\_Enable](#) [ThirdPartyFilter\\_RemoveFromList](#) [ThirdPartyFilter\\_ShowDialog](#)

Created with the Standard Edition of HelpNDoc: [Upgrade Your Documentation Process with a Help Authoring Tool](#)

## ThirdPartyFilter\_Enable

### TVideoGrabber.ThirdPartyFilter\_Enable

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Enables/disables a third-party filter.

#### Declaration

**function** ThirdPartyFilter\_Enable(Location: TThirdPartyFilterList; **Index:** LongInt; Enable: Boolean): Boolean;

**bool** \_\_fastcall ThirdPartyFilter\_Enable(TThirdPartyFilterList Location, **int** Index, **bool** Enable)

Function ThirdPartyFilter\_Enable(Location As TxThirdPartyFilterList, Index As Long, Enable As Boolean) As Boolean

#### Description

Used to enable/disable a third-party filter already inserted in a list.

**Location:** [TThirdPartyFilterList](#) list in which the filter has been added with [ThirdPartyFilter\\_AddToList](#) .

**index:** Index of the filter in the **Location** list. This index is the one returned by [ThirdPartyFilter\\_AddToList](#) .

**Enable:** enables or disables the filter.



#### See Also

[TThirdPartyFilterList](#) [TOnThirdPartyFilterConnected](#) [OnThirdPartyFilterConnected](#)  
[ThirdPartyFilter\\_AddToList](#) [ThirdPartyFilter\\_ClearList](#) [ThirdPartyFilter\\_RemoveFromList](#)  
[ThirdPartyFilter\\_ShowDialog](#)

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

### ThirdPartyFilter\_RemoveFromList

#### TVideoGrabber.ThirdPartyFilter\_RemoveFromList

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Removes a third-party filter from a list.

#### Declaration

**function** ThirdPartyFilter\_RemoveFromList(Location: TThirdPartyFilterList; **Index**: LongInt): Boolean;

**bool** \_\_fastcall ThirdPartyFilter\_RemoveFromList(TThirdPartyFilterList Location, **int** Index)

Function ThirdPartyFilter\_RemoveFromList(Location As TxThirdPartyFilterList, Index As Long) As Boolean

#### Description

Used to remove a third-party filter currently inserted in a list.

**Location**: [TThirdPartyFilterList](#) list in which the filter has been added with [ThirdPartyFilter\\_AddToList](#) .

**index**: Index of the filter in the **Location** list. This index is the one returned by [ThirdPartyFilter\\_AddToList](#) .

#### See Also

[TThirdPartyFilterList](#) [TOnThirdPartyFilterConnected](#) [OnThirdPartyFilterConnected](#)  
[ThirdPartyFilter\\_AddToList](#) [ThirdPartyFilter\\_ClearList](#) [ThirdPartyFilter\\_Enable](#) [ThirdPartyFilter\\_ShowDialog](#)

Created with the Standard Edition of HelpNDoc: [Free HTML Help documentation generator](#)

### ThirdPartyFilter\_ShowDialog

#### TVideoGrabber.ThirdPartyFilter\_ShowDialog

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Shows a third-party dialog.

#### Declaration

**function** ThirdPartyFilter\_ShowDialog(Location: TThirdPartyFilterList; **Index**: LongInt): Boolean;

**bool** \_\_fastcall ThirdPartyFilter\_ShowDialog(TThirdPartyFilterList Location, **int** Index)

Function ThirdPartyFilter\_ShowDialog(Location As TxThirdPartyFilterList, Index As Long) As Boolean

#### Description

Used to pop-up a third-party filter dialog, if available.

**Location**: [TThirdPartyFilterList](#) list in which the filter has been added with [ThirdPartyFilter\\_AddToList](#) .

**index**: Index of the filter in the **Location** list. This index is the one returned by [ThirdPartyFilter\\_AddToList](#) .

#### See Also

[TThirdPartyFilterList](#) [TOnThirdPartyFilterConnected](#) [OnThirdPartyFilterConnected](#)  
[ThirdPartyFilter\\_AddToList](#) [ThirdPartyFilter\\_ClearList](#) [ThirdPartyFilter\\_Enable](#)  
[ThirdPartyFilter\\_RemoveFromList](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## TVClearFrequencyOverrides

### TVideoGrabber.TVClearFrequencyOverrides

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Clears all the existing frequency overrides.

#### Declaration

**function** TVClearFrequencyOverrides: Boolean;

**bool** \_\_fastcall TVClearFrequencyOverrides(**void**)

Function TVClearFrequencyOverrides() As Boolean

#### Description

Used to clear all the existing frequency overrides set for a given country code and tuner input type by [TVSetChannelFrequencyOverride](#) .

**Important:** setting or clearing a frequency override will be applied only after restarting preview. This will be done automatically if [AutoRefreshPreview](#) is enabled.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

## TVGetMinMaxChannels

### TVideoGrabber.TVGetMinMaxChannels

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Retrieves the min and max TV channels for a given country code.

#### Declaration

**function** TVGetMinMaxChannels(**var** MinChannel: LongInt; **var** MaxChannel: LongInt): Boolean;

**bool** \_\_fastcall TVGetMinMaxChannels(**int** &MinChannel, **int** &MaxChannel)

Function TVGetMinMaxChannels(Min\_Channel, Max\_Channel) As Boolean

#### Description

Used to retrieve the min and max TV channels for a given country code.  
Be sure to select the proper [country code](#) before invoking this function.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation](#)

## TVSetChannelFrequencyOverride

### TVideoGrabber.TVSetChannelFrequencyOverride

[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Sets a frequency overrides.

#### Declaration

**function** TVSetChannelFrequencyOverride(TVChannel: LongInt; FrequencyInHz: LongInt): LongInt;

**int** \_\_fastcall TVSetChannelFrequencyOverride(**int** TVChannel, **int** FrequencyInHz)

Function TVSetChannelFrequencyOverride(TVChannel As Long, FrequencyInHz As Long) As Long

#### Description

Used to set a frequency override for a given TV channel, for the current [country code](#) and [tuner input type](#).

This function sets a frequency override on the specified TV channel.

This will affect the TV channel for the current [country code](#) and [tuner input type](#) only.

If you need to retrieve the default frequency for this TV channel, simply assign the channel number to the [TVChannel](#) property and the [OnTVChannelSelected](#) event will return the frequency (the event is synchrone).

To erase any frequency override on this TV channel simply pass " -1" as frequency value in the FrequencyInHz parameter.

**Important:** setting or clearing a frequency override will be applied only after restarting preview. This will be done automatically if [AutoRefreshPreview](#) is enabled.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## TVStartAutoScan

### TVideoGrabber.TVStartAutoScan

[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Starts the automatic scanning of TV channels.

#### Declaration

**function** TVStartAutoScan: Boolean;

**bool** \_\_fastcall TVStartAutoScan(**void**)

Function TVStartAutoScan() As Boolean

#### Description

Used to start the automatic scanning of TV channels for a given country code and tuner input type.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## TVStopAutoScan

### TVideoGrabber.TVStopAutoScan

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stops the automatic scanning of TV channels.

#### Declaration

**function** TVStopAutoScan: Boolean;

**bool** \_\_fastcall TVStopAutoScan(**void**)

Function TVStopAutoScan() As Boolean

#### Description

Used to stop the automatic scanning of TV channels, previously started by invoking [TVStartAutoScan](#) .

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's User-Friendly UI](#)

## UseNearestVideoSize

### TVideoGrabber.UseNearestVideoSize

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Automatically selects the nearest video source size to a preferred width and height.

#### Declaration

**procedure** UseNearestVideoSize(PreferredWidth, PreferredHeight: LongInt; Stretch: Boolean);

**void** \_\_fastcall UseNearestVideoSize(**int** PreferredWidth, **int** PreferredHeight, **bool** Stretch)

Sub UseNearestVideoSize(PreferredWidth As Long, PreferredHeight As Long, Stretch As Boolean)

#### Description

Used to automatically use the video capture device size nearest to a preferred width and height.

- if **stretch** disabled, the control is resized to the nearest size found.
- if **stretch** enabled, the nearest size is selected, the control is resized to the preferred width and height,

and the video window is stretched to fit the control.

E.g. we wish a 320x240 display size, and the video capture device offers only 176x144 and 352x288 sizes.

- UseNearestVideoSize (320, 240, false) selects the 352x288 video size and resize the display window to 352x288.
- UseNearestVideoSize (320, 240, true) selects the 352x288 video size and stretches the display window to 320x240.

**Important note:** when this function is called, the [VideoSize](#) property is ignored. To re-enable the [VideoSize](#) property, call UseNearestVideoSize (0, 0, false).

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [VideoFormat](#) [VideoFormatIndex](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

---

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

---

## v360\_AddYawPitchRoll

### TVideoGrabber.v360\_AddYawPitchRoll

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: modifies the point of view

#### Declaration

**function** v360\_AddYawPitchRoll (Yaw: Double; Pitch: Double; Roll: Double): Boolean;

**bool** v360\_AddYawPitchRoll(**double** Yaw, **double** Pitch, **double** Roll);

#### Description

Used to increase or decrease the yaw, pitch or roll, in the -180.0 to 180.0 range.

E.g.:

v360\_AddYawPitchRoll (-10.0, 5.5, 0.0);

#### See Also

[v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#) [v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

---

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

---

## v360\_GetAngle

### TVideoGrabber.v360\_GetAngle

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: retrieve an angle

#### Declaration

**function** v360\_GetAngle (Direction: Tv360\_InOut; Angle: Tv360\_Angle): Double;

**double** v360\_GetAngle(Tv360\_InOut Direction, Tv360\_Angle Angle);

#### Description

Used to retrieve the current value of the input or output angles (vertical, horizontal or diagonal)

- **1st parameter:** [Tv360\\_InOut](#) Direction specifies the direction
- **2nd parameter:** [Tv360\\_Angle](#) Angle specify what angle must be set  
The value returned is in the 0..180 range.

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetYawPitchRoll](#) [v360\\_MasterAngle](#)  
[v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#)  
[v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#)  
[v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## v360\_GetYawPitchRoll

### TVideoGrabber.v360\_GetYawPitchRoll

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: retrieves the point of view

#### Declaration

**procedure** v360\_GetYawPitchRoll (**var** Yaw: Double; **var** Pitch: Double; **var** Roll: Double);

**void** v360\_GetYawPitchRoll(**double** &Yaw, **double** &Pitch, **double** &Roll);

#### Description

Used to retrieve the current point of view (yaw, pitch and roll)

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_MasterAngle](#)  
[v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#) [v360\\_SetAngle](#)  
[v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#)  
[v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Protect Your Confidential PDFs with These Simple Security Measures](#)

## v360\_ResetAnglesToDefault

### TVideoGrabber.v360\_ResetAnglesToDefault

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: reset the angles

#### Declaration

**procedure** v360\_ResetAnglesToDefault();

**void** v360\_ResetAnglesToDefault(**void**);

#### Description

Resets the angles to their default values

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)  
[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_SetAngle](#) [v360\\_SetInterpolation](#)  
[v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## v360\_SetAngle

### TVideoGrabber.v360\_SetAngle

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: set an angle

#### Declaration

**function** v360\_SetAngle (Direction: Tv360\_InOut; Angle: Tv360\_Angle; Value: Double): Boolean;

bool v360\_SetAngle(Tv360\_InOut Direction, Tv360\_Angle Angle, double Value);

#### Description

Used to set the input or output angles (vertical, horizontal or diagonal)

- **1st parameter:** [Tv360\\_InOut](#) Direction specifies the direction
- **2nd parameter:** [Tv360\\_Angle](#) Angle specify what angle must be set
- **3rd parameter:** the value of the angle in the 0..180 range.

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)  
[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#)  
[v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#)  
[v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## v360\_SetInterpolation

### TVideoGrabber.v360\_SetInterpolation

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: sets the interpolation method

#### Declaration

**function** v360\_SetInterpolation (Value: Tv360\_Interpolation): Boolean;

bool v360\_SetInterpolation(Tv360\_Interpolation Value);

#### Description

Parameter of type [Tv360\\_Interpolation](#)

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)  
[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#)  
[v360\\_SetAngle](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## v360\_SetProjection

### TVideoGrabber.v360\_SetProjection

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: sets the type of projection

#### Declaration

**function** v360\_SetProjection (Direction: Tv360\_InOut; Value: Tv360\_Projection): Boolean;



**bool** v360\_SetProjection(Tv360\_InOut Direction, Tv360\_Projection Value);

#### Description

##### Parameters:

Direction: [Tv360\\_InOut](#);

Value: [Tv360\\_Projection](#)

##### The default value are:

input: ipp\_Equirectangular

output: ipp\_Flat

##### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)

[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#)

[v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc

## v360\_SetStereoFormat

### TVideoGrabber.v360\_SetStereoFormat

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: sets the stereo format

#### Declaration

**function** v360\_SetStereoFormat (Direction: Tv360\_InOut; Value: Tv360\_StereoFormat): Boolean;

**bool** v360\_SetStereoFormat(Tv360\_InOut Direction, Tv360\_StereoFormat Value);

#### Description

Used to set the input and/or output stereo format

##### Parameters:

Direction: [Tv360\\_InOut](#)

Value: [Tv360\\_StereoFormat](#)

##### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)

[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#)

[v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetTranspose](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature

## v360\_SetTranspose

### TVideoGrabber.v360\_SetTranspose

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: enables/disables the transposition

#### Declaration

**function** v360\_SetTranspose (Direction: Tv360\_InOut; Value: Boolean): Boolean;

**bool** v360\_SetTranspose(Tv360\_InOut Direction, **bool** Value);

#### Description

Parameters:

Direction: [Tv360\\_InOut](#)

Value: Boolean

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)  
[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#)  
[v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetYawPitchRoll](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Edit and Export Markdown Documents](#)

## v360\_SetYawPitchRoll

### TVideoGrabber.v360\_SetYawPitchRoll

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

360° video: sets the point of view

#### Declaration

**function** v360\_SetYawPitchRoll (Yaw: Double; Pitch: Double; Roll: Double): Boolean;

**bool** v360\_SetYawPitchRoll(**double** Yaw, **double** Pitch, **double** Roll);

#### Description

Used to set the yaw, pitch or roll, in the -180.0 to 180.0 range.

E.g.:

v360\_SetYawPitchRoll (12.0, 5.0, 0.0);

#### See Also

[v360\\_AddYawPitchRoll](#) [v360\\_AspectRatio](#) [v360\\_Enabled](#) [v360\\_GetAngle](#) [v360\\_GetYawPitchRoll](#)  
[v360\\_MasterAngle](#) [v360\\_MouseAction](#) [v360\\_MouseActionPercent](#) [v360\\_ResetAnglesToDefault](#)  
[v360\\_SetAngle](#) [v360\\_SetInterpolation](#) [v360\\_SetProjection](#) [v360\\_SetStereoFormat](#) [v360\\_SetTranspose](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## VideoCompressorIndex

### TVideoGrabber.VideoCompressorIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given compressor in the [VideoCompressors](#) list.

#### Declaration

**function** VideoCompressorIndex(Value: **String**): LongInt;

**int** \_\_fastcall VideoCompressorIndex(wchar\_t \*Value)

Function VideoCompressorIndex(param1 As String) As Long

#### Description

Used to retrieve the index of a given compressor in the [VideoCompressors](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.VideoCompressor := VideoCompressorIndex ('Microsoft MPEG-4 Video Codec V2')
VideoGrabber1.VideoCompressor := VideoCompressorIndex ('Microsoft MPEG-4*');           // s
VideoGrabber1.VideoCompressor := VideoCompressorIndex ('*MPEG-4*');                   // selects the
```

**See Also**

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: [Easily share your documentation with the world through a beautiful website](#)

**VideoDeviceIndex****TVideoGrabber.VideoDeviceIndex**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given audio device in the [VideoDevices](#) list.

**Declaration**

**function** VideoDeviceIndex(Value: **String**): LongInt;

**int** \_\_fastcall VideoDeviceIndex(wchar\_t \*Value)

Function VideoDeviceIndex(param1 As String) As Long

**Description**

Used to retrieve the index of a given video device in the [VideoDevices](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.VideoDevice := VideoDeviceIndex ('Microsoft DV Camera and VCR (WDM)'); // s
VideoGrabber1.VideoDevice := VideoDeviceIndex ('Microsoft DV Camera*');           // s
VideoGrabber1.VideoDevice := VideoDeviceIndex ('*DV Camera*');                   // s
```

**See Also**

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

**VideoDeviceIndexFromId****TVideoGrabber.VideoDeviceIndexFromId**[Prev](#)[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given video device in the [VideoDevicesId](#) list

**Declaration**

**function** VideoDeviceIndexFromId(Value: **String**): LongInt;

```
int __fastcall VideoDeviceIndexFromId(wchar_t *Value)
```

Function VideoDeviceIndexFromId(param1 As String) As Long

### Description

Used to retrieve the index of a given video device in the [VideoDevicesId](#) list.  
This lets select the device by its hardware identifier rather than its manufacturer name.

E.g.:

```
VideoGrabber1.VideoDevice := VideoDeviceIndexFromId
('devicepnpusbvid046dpid0826mi02739103c9a0000265e8773d-8f56-11d0-a3b9-00a0c9223196globalHD Webcam C525');
```

### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## VideoFormatIndex

### TVideoGrabber.VideoFormatIndex

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

index of a video format

### Declaration

```
function VideoFormatIndex(Value: String): LongInt;
```

```
int __fastcall VideoFormatIndex(wchar_t *Value)
```

Function VideoFormatIndex(Value As String) As Long

### Description

Used to retrieve the index of a given video format in the [VideoFormats](#) list

### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#) [GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoSize](#) [VideoSizeIndex](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## VideoFromImages\_CreateSetOfBitmaps

### TVideoGrabber.VideoFromImages\_CreateSetOfBitmaps

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Methods](#)

Creates a set of bitmaps that will be used to create a video clip.

### Declaration

```
function VideoFromImages_CreateSetOfBitmaps: Boolean;
```

**bool** \_\_fastcall VideoFromImages\_CreateSetOfBitmaps(**void**)

Function VideoFromImages\_CreateSetOfBitmaps() As Boolean

### Description

Creates a set of bitmaps stored in a temporary file, that will be used to create a video clip from bitmaps.  
See [Video clip from bitmaps overview](#).

Created with the Standard Edition of HelpNDoc: [Keep Your PDFs Safe from Unauthorized Access with These Security Measures](#)

## VideoInputIndex

### TVideoGrabber.VideoInputIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given audio input in the [VideoInputs](#) list.

### Declaration

**function** VideoInputIndex(Value: **String**): LongInt;

**int** \_\_fastcall VideoInputIndex(wchar\_t \*Value)

Function VideoInputIndex(param1 As String) As Long

### Description

Used to retrieve the index of a given audio input in the [VideoInputs](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.VideoInput := VideoInputIndex ('SVideo'); // selects the "SVideo" video input
VideoGrabber1.VideoInput := VideoInputIndex ('SVideo*'); // selects the 1st video input
VideoGrabber1.VideoInput := VideoInputIndex (*SVideo*); // selects the 1st video input
```

### See Also

[Video inputs](#) [VideoInput](#) [VideoInputs](#) [VideoInputsCount](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## VideoQualityAuto

### TVideoGrabber.VideoQualityAuto

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Reports if a [TVideoQuality](#) setting is in "auto" mode.

### Declaration

**function** GetVideoQualityAuto(Setting: TVideoQuality): Boolean;

**bool** \_\_fastcall GetVideoQualityAuto(TVideoQuality Setting)

Function GetVideoQualityAuto(Setting As TVideoQuality) As Boolean

### Description

Used to know if a [TVideoQuality](#) setting is in "auto" mode or "manual" mode.

Returns "true" if the setting is in "auto" mode, "false" if it is in "manual" mode.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## VideoQualityDefault

### TVideoGrabber.VideoQualityDefault

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Resets a [TVideoQuality](#) setting.

#### Declaration

**function** VideoQualityDefault(Setting: TVideoQuality): LongInt;

**int** \_\_fastcall VideoQualityDefault(TVideoQuality Setting)

Function VideoQualityDefault(Setting As TxVideoQuality) As Long

#### Description

Used to reset a [TVideoQuality](#) setting to its default value.  
Returns MAXINT upon failure.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## VideoQualityMax

### TVideoGrabber.VideoQualityMax

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Max value of a [TVideoQuality](#) setting.

#### Declaration

**function** VideoQualityMax(Setting: TVideoQuality): LongInt;

**int** \_\_fastcall VideoQualityMax(TVideoQuality Setting)

Function VideoQualityMax(Setting As TxVideoQuality) As Long

#### Description

Used to retrieve the max value of a [TVideoQuality](#) setting.  
Returns MAXINT upon failure.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#) [VideoQualityDefault](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Import and export Markdown documents](#)

## VideoQualityMin

### TVideoGrabber.VideoQualityMin

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Min value of a [TVideoQuality](#) setting.

#### Declaration

**function** VideoQualityMin(Setting: TVideoQuality): LongInt;

**int** \_\_fastcall VideoQualityMin(TVideoQuality Setting)

Function VideoQualityMin(Setting As TxVideoQuality) As Long

#### Description

Used to retrieve the min value of a [TVideoQuality](#) setting.  
Returns MAXINT upon failure.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#)  
[VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualitySettings](#) [VideoQualityStep](#) [VideoRendererPriority](#)  
[TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

## VideoQualityStep

### TVideoGrabber.VideoQualityStep

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Stepping value of a [TVideoQuality](#) setting.

#### Declaration

**function** VideoQualityStep(Setting: TVideoQuality): LongInt;

**int** \_\_fastcall VideoQualityStep(TVideoQuality Setting)

Function VideoQualityStep(Setting As TxVideoQuality) As Long

#### Description

Used to retrieve the stepping value of a [TVideoQuality](#) setting.  
Returns MAXINT upon failure.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#)  
[VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoRendererPriority](#)  
[TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## VideoQualityValue

### TVideoGrabber.VideoQualityValue

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)



Gets a [TVideoQuality](#) current value.

#### Declaration

**function** GetVideoQuality(Setting: TVideoQuality): LongInt;

**int** \_\_fastcall GetVideoQuality(TVideoQuality Setting)

Function GetVideoQuality(Setting As TVideoQuality) As Long

#### Description

Used to retrieves the current value of a [TVideoQuality](#) setting.  
Returns MAXINT upon failure.

#### See Also

[IsVideoQualityAvailable](#) [IsVideoQualitySettingAvailable](#) [SetVideoQuality](#) [VideoQualityAuto](#)  
[VideoQualityDefault](#) [VideoQualityMax](#) [VideoQualityMin](#) [VideoQualitySettings](#) [VideoQualityStep](#)  
[VideoRendererPriority](#) [TVideoQuality](#)

Created with the Standard Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

## VideoSizeIndex

### TVideoGrabber.VideoSizeIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given audio input in the [VideoSizes](#) list.

#### Declaration

**function** VideoSizeIndex(Value: **String**): LongInt;

**int** \_\_fastcall VideoSizeIndex(wchar\_t \*Value)

Function VideoSizeIndex(param1 As String) As Long

#### Description

Used to retrieve the index of a given audio input in the [VideoSizes](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.Videosize := VideosizeIndex ('720x576'); // selects the "720x576" video size
VideoGrabber1.Videosize := VideosizeIndex ('720x*');    // selects the 1st video size t
VideoGrabber1.Videosize := VideosizeIndex (*720x*');    // selects the 1st video size t
```

#### See Also

[GetNearestVideoHeight](#) [GetNearestVideoSize](#) [GetNearestVideoWidth](#) [GetVideoHeightFromIndex](#)  
[GetVideoSizeFromIndex](#) [GetVideoWidthFromIndex](#) [UseNearestVideoSize](#) [VideoFormat](#) [VideoFormatIndex](#)  
[VideoSize](#) [VideoSizes](#) [VideoSizesCount](#)

Created with the Standard Edition of HelpNDoc: [Streamline your documentation process with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## VideoSubtypeIndex

### TVideoGrabber.VideoSubtypeIndex

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Methods](#)

Returns the index of a given audio input in the [VideoSubtypes](#) list.

#### Declaration

**function** VideoSubtypeIndex(Value: **String**): LongInt;

**int** \_\_fastcall VideoSubtypeIndex(wchar\_t \*Value)

Function VideoSubtypeIndex(param1 As String) As Long

#### Description

Used to retrieve the index of a given audio input in the [VideoSubtypes](#) list.

The function accepts wild chars. E.g.:

```
VideoGrabber1.VideoSubtype := VideoSubtypeIndex ('RGB24'); // selects the "RGB24" video sub
VideoGrabber1.VideoSubtype := VideoSubtypeIndex ('RGB*'); // selects the 1st video sub
VideoGrabber1.VideoSubtype := VideoSubtypeIndex ('*RGB*'); // selects the 1st video sub
```

#### See Also

[VideoSubtype](#) [VideoSubtypes](#) [VideoSubtypesCount](#)

Created with the Standard Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

## WriteScriptCommand

### TVideoGrabber.WriteScriptCommand

[Prev](#)

[TVideoGrabber](#) [Methods](#)

Not implemented

#### Declaration

**function** WriteScriptCommand (ScriptType: **string**; ScriptArgument: **string**; ScriptTime: int64): Boolean;

**bool** \_\_fastcall WriteScriptCommand (wchar\_t \*ScriptType, wchar\_t \*ScriptArgument, \_\_int64 ScriptTime) ;

function WriteScriptCommand (ScriptType as String, ScriptArgument as String, ScriptTime as Double) as Boolean

#### Description

Not implemented

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

## Events

### TVideoGrabber Events

[TVideoGrabber](#)

#### Public

[OnAudioBufferNegotiation](#)  
[OnAudioDeviceSelected](#)  
[OnAudioPeak](#)  
[OnAuthenticationNeeded](#)  
[OnAVIDurationUpdated](#)

[OnBacktimedFramesCountReached](#)  
[OnBitmapsLoadingProgress](#)  
[OnClick](#)  
[OnClientConnection](#)  
[OnColorKeyChange](#)  
[OnCopyPreallocDataCompleted](#)  
[OnCopyPreallocDataProgress](#)  
[OnCopyPreallocDataStarted](#)  
[OnCreatePreallocFileCompleted](#)  
[OnCreatePreallocFileProgress](#)  
[OnCreatePreallocFileStarted](#)  
[OnDbClick](#)  
[OnDeviceArrivalOrRemoval](#)  
[OnDeviceLost](#)  
[OnDeviceReconnected](#)  
[OnDeviceReconnecting](#)  
[OnDirectNetworkStreamingHostUrl](#)  
[OnDiskFull](#)  
[OnDragDropFiles](#)  
[OnDVCommandCompleted](#)  
[OnDVDDiscontinuity](#)  
[OnEnumerateWindows](#)  
[OnFilterSelected](#)  
[OnFrameBitmap](#)  
[OnFrameCaptureCompleted](#)  
[OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEONHDR](#)  
[OnFrameProgress](#)  
[OnFrameProgress2](#)  
[OnGraphBuilt](#)  
[OnInactive](#)  
[OnKeyPress](#)  
[OnLastCommandCompleted](#)  
[OnLeavingFullScreen](#)  
[OnLog](#)  
[OnMotionDetected](#)  
[OnMotionNotDetected](#)  
[OnMouseDown](#)  
[OnMouseDown\\_Video](#)  
[OnMouseDown\\_Window](#)  
[OnMouseMove](#)  
[OnMouseMove\\_Video](#)  
[OnMouseMove\\_Window](#)  
[OnMouseUp](#)  
[OnMouseUp\\_Video](#)  
[OnMouseUp\\_Window](#)  
[OnMouseWheel](#)  
[OnNoVideoDevices](#)  
[OnNTPTimeStamp](#)  
[OnONVIFDiscoveryCompletedNotification](#)  
[OnOpenURLAsyncStatusChanged](#)  
[OnPlayerBufferingData](#)  
[OnPlayerEndOfPlaylist](#)  
[OnPlayerEndOfStream](#)  
[OnPlayerOpened](#)  
[OnPlayerStateChanged](#)  
[OnPlayerUpdateTrackbarPosition](#)  
[OnPreviewStarted](#)  
[OnRawAudioSample](#)  
[OnRawVideoSample](#)  
[OnRecordingCompleted](#)  
[OnRecordingPaused](#)

[OnRecordingReadyToStart](#)  
[OnRecordingStarted](#)  
[OnReencodingCompleted](#)  
[OnReencodingProgress](#)  
[OnReencodingStarted](#)  
[OnReinitializing](#)  
[OnResizeVideo](#)  
[OnTextOverlayScrollingCompleted](#)  
[OnThirdPartyFilterConnected](#)  
[OnThreadSync](#)  
[OnTVChannelScanCompleted](#)  
[OnTVChannelScanStarted](#)  
[OnTVChannelSelected](#)  
[OnVideoCompressionSettings](#)  
[OnVideoDeviceSelected](#)  
[OnVideoFromBitmapsNextFrameNeeded](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Encrypted, Password-Protected PDFs](#)

## OnAudioBufferNegotiation

### TVideoGrabber.OnAudioBufferNegotiation

[Next](#)

[TVideoGrabber](#) [Events](#)

Lets choose the audio buffer size when rendering the audio stream

#### Declaration

**property** OnAudioBufferNegotiation: TEventNotification **read** FOnAudioBufferNegotiation **write** FOnAudioBufferNegotiation;

**\_\_property** TOnAudioBufferNegotiation OnAudioBufferNegotiation=read=FOnAudioBufferNegotiation, write=FOnAudioBufferNegotiation

Event OnAudioBufferNegotiation

#### Description

This event occurs when rendering the audio stream, allowing to change the default audio buffer size, by modifying the ProposedBufferSize parameter from this event

Type: [TOnAudioBufferNegotiation](#)

#### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#)  
[AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#)  
[AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#)  
[IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#)  
[IsAudioInputBalanceAvailable](#) [OnAudioDeviceSelected](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## OnAudioDeviceSelected

### TVideoGrabber.OnAudioDeviceSelected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when an audio capture device is selected.

#### Declaration

**property** OnAudioDeviceSelected: TEventNotification **read** FOnAudioDeviceSelected **write** FOnAudioDeviceSelected;

\_\_property TEventNotification OnAudioDeviceSelected=read=FOnAudioDeviceSelected, write=FOnAudioDeviceSelected

Event OnAudioDeviceSelected()

### Description

Occurs when an audio capture device is selected by the [AudioDevice](#) property. The device-dependent settings of this audio capture device are reloaded and this event occurs.

### See Also

[TAudioFormat](#) [AssociateAudioAndVideoDevices](#) [AudioBalance](#) [AudioDevice](#) [AudioDeviceIndex](#) [AudioDeviceName](#) [AudioDeviceRendering](#) [AudioDevices](#) [AudioDevicesCount](#) [AudioFormat](#) [AudioFormats](#) [AudioInput](#) [AudioInputIndex](#) [AudioInputLevel](#) [AudioInputMono](#) [AudioInputs](#) [AudioInputsCount](#) [AudioSource](#) [IsAudioCrossbarAvailable](#) [IsAudioDeviceASoundCard](#) [IsAudioDeviceConnected](#) [IsAudioInputBalanceAvailable](#) [OnAudioBufferNegotiation](#) [SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

## OnAudioPeak

### TVideoGrabber.OnAudioPeak

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Returns the audio peak levels, in percentage or in DB.

### Declaration

**property** OnAudioPeak: TOnAudioPeak **read** FOnAudioPeak **write** FOnAudioPeak;

\_\_property TOnAudioPeak OnAudioPeak=read=FOnAudioPeak, write=FOnAudioPeak

Event OnAudioPeak()

### Description

This [TOnAudioPeak](#) event returns the audio peak levels for the left and right audio channels, in percentage and in DB.

The [AudioPeakEvent](#) property must be enabled to activate this event.

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Creation with a Help Authoring Tool](#)

## OnAuthenticationNeeded

### TVideoGrabber.OnAuthenticationNeeded

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs to pass a user name and password when an authentication is required

### Declaration

**property** OnAuthenticationNeeded: TOnAuthenticationNeeded **read** FOnAuthenticationNeeded **write** FOnAuthenticationNeeded;

\_\_property TOnAuthenticationNeeded OnAuthenticationNeeded=read=FOnAuthenticationNeeded, write=OnAuthenticationNeeded

Event OnAuthenticationNeeded(Id As Long, Realm as String, Server as String, UserName, Password)

## Description

This [TOnAuthenticationNeeded](#) event occurs when an authentication is needed, e.g. to send streaming to a publishing point.

Parameters:

**Id:** reserved

**Realm:** reports the domain name, if any

**Server:** reports the server name, if any

**UserName:** set this property with the user name needed for the authentication

**Password:** set this parameter with the password needed for the authentication

## See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#)  
[ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

## OnAVIDurationUpdated

### TVideoGrabber.OnAVIDurationUpdated

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the AVI recording ends and the [AVIDurationUpdated](#) property is enabled.

## Declaration

**property** OnAVIDurationUpdated: TOnAVIDurationUpdated **read** FOnAVIDurationUpdated **write** FOnAVIDurationUpdated;

\_\_property TOnAVIDurationUpdated OnAVIDurationUpdated=read=FOnAVIDurationUpdated, write=FOnAVIDurationUpdated

Event OnAVIDurationUpdated(NewStateValue As Long, NewStateLabel As String)

## Description

Occurs when closing an AVI file just recorded and the [AVIDurationUpdated](#) property is enabled.

This [TOnAVIDurationUpdated](#) event lets you update the frame rate and/or duration of the AVI clip, if needed.

This event is designed only to give access to the frame rate and duration of the AVI. **Be sure not to use it as notification for the end of recording.**

The end of the recording is notified by the [OnRecordingCompleted](#) event, whatever the kind of recording (AVI, ASF, MPEG).

Parameters:

**FileName:** returns name of the file recorded,

**Frames:** returns the number of frames recorded,

**FrameRate:** returns the real frame rate (can be adjusted),

**Duration:** returns the duration of the recording (can be adjusted).

## See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## OnBacktimedFramesCountReached

### TVideoGrabber.OnBacktimedFramesCountReached

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs the when [AVIBacktimedVideoFramesCount](#) is enabled (> 0) and back-timed video frames start replacing normal video frames in the video stream.

#### Declaration

**property** OnBacktimedFramesCountReached: TEventNotification **read**  
FOnBacktimedFramesCountReached **write** FOnBacktimedFramesCountReached;

\_\_property TEventNotification  
OnBacktimedFramesCountReached=read=FOnBacktimedFramesCountReached,  
write=FOnBacktimedFramesCountReached

Event OnBacktimedFramesCountReached()

#### Description

Occurs the when [AVIBacktimedVideoFramesCount](#) is enabled (> 0) and back-timed video frames start replacing normal video frames in the video stream.

See [AVIBacktimedVideoFramesCount](#) for more information.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector](#) [CellMotionRatio](#) [MotionDetector](#) [CompareBlue](#) [MotionDetector](#) [CompareGreen](#) [MotionDetector](#) [CompareRed](#) [MotionDetector](#) [Enabled](#) [MotionDetector](#) [EnumGridDialogControls](#) [MotionDetector](#) [Get2DTextGrid](#) [MotionDetector](#) [Get2DTextMotion](#) [MotionDetector](#) [GetCellLocation](#) [MotionDetector](#) [GetCellSensitivity](#) [MotionDetector](#) [GetCellSize](#) [MotionDetector](#) [GloballyIncOrDecSensitivity](#) [MotionDetector](#) [GlobalMotionRatio](#) [MotionDetector](#) [GreyScale](#) [MotionDetector](#) [Grid](#) [MotionDetector](#) [GridXCount](#) [MotionDetector](#) [GridYCount](#) [MotionDetector](#) [IsGridValid](#) [MotionDetector](#) [MaxDetectionsPerSecond](#) [MotionDetector](#) [ReduceCPULoad](#) [MotionDetector](#) [ReduceVideoNoise](#) [MotionDetector](#) [Reset](#) [MotionDetector](#) [ResetGlobalSensitivity](#) [MotionDetector](#) [SetCellSensitivity](#) [MotionDetector](#) [SetGridSize](#) [MotionDetector](#) [ShowGridDialog](#) [MotionDetector](#) [Triggered](#) [MotionDetector](#) [UseThisReferenceSample](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion](#) [Enabled](#) [RecordingOnMotion](#) [MotionThreshold](#) [RecordingOnMotion](#) [NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Documents to eBooks: A Step-by-Step Guide with HelpNDoc](#)



## OnBitmapsLoadingProgress

### TVideoGrabber.OnBitmapsLoadingProgress

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs periodically when building a set of bitmap that will be used to create a video clip.

#### Declaration

**property** OnBitmapsLoadingProgress: TOnProgressCommented **read** FOnBitmapsLoadingProgress **write** FOnBitmapsLoadingProgress;

\_\_property TOnProgressCommented OnBitmapsLoadingProgress=read=FOnBitmapsLoadingProgress, write=FOnBitmapsLoadingProgress

Event OnBitmapsLoadingProgress(Percent As Long, Position As Double, Duration As Double, Comment As String)

#### Description

This [TOnProgressCommented](#) event occurs periodically when invoking [VideoFromImages](#) [CreateSetOfBitmaps](#) to build a set of bitmap that will be used to create a video clip.

#### See Also

[SendImageToVideoFromBitmaps](#) [SendImageToVideoFromBitmaps2](#) [VideoFromImages](#) [BitmapsSortedByVideoFromImages](#) [RepeatIndefinitely](#) [VideoFromImages](#) [SourceDirectory](#) [VideoFromImages](#) [TemporaryFile](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Documents to eBooks: A Step-by-Step Guide with HelpNDoc](#)

## OnClick

### TVideoGrabber.OnClick

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

normal OnClick event

#### Declaration

**property** OnClick: TEventNotification **read** FOnClick **write** FOnClick;

\_\_property TEventNotification OnClick=read=FOnClick, write=FOnClick

Event OnClick()

#### Description

normal OnClick event

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

## OnClientConnection

### TVideoGrabber.OnClientConnection

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a network client connects or disconnects

## Declaration

**property** OnClientConnection: TOnClientConnection **read** FOnClientConnection **write** FOnClientConnection;

\_\_property TOnClientConnection OnClientConnection=read=FOnClientConnection, write=FOnClientConnection

Event OnClientConnection (Connected as Boolean, ClientInfo as String)

## Description

This [TOnClientConnection](#) event occurs when the streaming is running (NetworkStreaming <> ns\_Disabled), and a client connects or disconnects.

## Parameters

**Connected:** connection = true, disconnection = false

**ClientInfo:** returns the client information (usually IP:port)

## See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

## OnColorKeyChange

### TVideoGrabber.OnColorKeyChange

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Returns the color key for transparent windows.

## Declaration

**property** OnColorKeyChange: TOnColorKeyChange **read** FOnColorKeyChange **write** FOnColorKeyChange;

\_\_property TOnColorKeyChange OnColorKeyChange=read=FOnColorKeyChange, write=FOnColorKeyChange

Event OnColorKeyChange(ColorKey As OLE\_COLOR)

## Description

When [ColorKeyEnabled](#) is enabled, this [TOnColorKeyChange](#) event occurs when starting preview, recording or playback.

This event returns the color key to use for transparent windows.

See the *VideoGrabber1ColorKeyChange* event in the *MainDemo* project for sample code.

## See Also

[TOnColorKeyChange](#) [ColorKeyEnabled](#)

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

## OnCopyPreallocDataCompleted

### TVideoGrabber.OnCopyPreallocDataCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs after the preallocated recording file has been fully copied.

#### Declaration

**property** OnCopyPreallocDataCompleted: TOnSourceFileToDestFileCompleted **read** FOnCopyPreallocDataCompleted **write** FOnCopyPreallocDataCompleted;

```
__property TOnSourceFileToDestFileCompleted
OnCopyPreallocDataCompleted=read=FOnCopyPreallocDataCompleted,
write=FOnCopyPreallocDataCompleted
```

Event OnCopyPreallocDataCompleted(SourceFile As String, DestFile As String, Success As Boolean)

#### Description

This [TOnSourceFileToDestFileCompleted](#) event is used when [PreallocCapFileEnabled](#) is true (a preallocated huge capture file is used for recording).  
Occurs when the streams data has been fully copied from the pre-allocated AVI file to the final AVI file.  
As a result, the new file can be much smaller than the preallocated file.

#### See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#)  
[OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileProgress](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

## OnCopyPreallocDataProgress

### TVideoGrabber.OnCopyPreallocDataProgress

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs periodically during the preallocated file copy.

#### Declaration

**property** OnCopyPreallocDataProgress: TOnProgress **read** FOnCopyPreallocDataProgress **write** FOnCopyPreallocDataProgress;

```
__property TOnProgress OnCopyPreallocDataProgress=read=FOnCopyPreallocDataProgress,
write=FOnCopyPreallocDataProgress
```

Event OnCopyPreallocDataProgress(Percent As Long, Position As Double, Duration As Double)

#### Description

This [TOnProgress](#) event occurs periodically during the preallocated file copy.  
Used when [PreallocCapFileEnabled](#) is true (a preallocated huge capture file is used for recording).  
Occurs when the streams data has been fully copied from the pre-allocated AVI file to the final AVI file.

#### See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataStarted](#)  
[OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileProgress](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## OnCopyPreallocDataStarted

### TVideoGrabber.OnCopyPreallocDataStarted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the preallocated recording file begins to be copied.

#### Declaration

**property** OnCopyPreallocDataStarted: TOnSourceFileToDestFileStarted **read**  
FOnCopyPreallocDataStarted **write** FOnCopyPreallocDataStarted;

\_\_property TOnSourceFileToDestFileStarted  
OnCopyPreallocDataStarted=read=FOnCopyPreallocDataStarted, write=FOnCopyPreallocDataStarted

Event OnCopyPreallocDataStarted(SourceFile As String, DestFile As String)

#### Description

This [TOnSourceFileToDestFileStarted](#) event is used when [PreallocCapFileEnabled](#) is true (a preallocated huge capture file is used for recording).

Occurs when the streams data begins to be copied from the pre-allocated AVI file to the final AVI file.

#### See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)  
[OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileProgress](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## OnCreatePreallocFileCompleted

### TVideoGrabber.OnCreatePreallocFileCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the preallocated recording file has been fully created.

#### Declaration

**property** OnCreatePreallocFileCompleted: TOnCreatePreallocatedFileCompleted **read**  
FOnCreatePreallocFileCompleted **write** FOnCreatePreallocFileCompleted;

\_\_property TOnCreatePreallocatedFileCompleted  
OnCreatePreallocFileCompleted=read=FOnCreatePreallocFileCompleted,  
write=FOnCreatePreallocFileCompleted

Event OnCreatePreallocFileCompleted(FileName As String, Success As Boolean)

#### Description

This [TOnCreatePreallocatedFileCompleted](#) event is used when [PreallocCapFileEnabled](#) is true (a preallocated huge capture file is used for recording).

Occurs when the preallocated recording has been fully created.

#### See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)  
[OnCopyPreallocDataStarted](#) [OnCreatePreallocFileProgress](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize your documentation process with HelpNDoc's online capabilities](#)

## OnCreatePreallocFileProgress

### TVideoGrabber.OnCreatePreallocFileProgress

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs periodically during the creation of the preallocated file.

#### Declaration

**property** OnCreatePreallocFileProgress: TOnProgress **read** FOnCreatePreallocFileProgress **write** FOnCreatePreallocFileProgress;

\_\_property TOnProgress OnCreatePreallocFileProgress=read=FOnCreatePreallocFileProgress, write=FOnCreatePreallocFileProgress

Event OnCreatePreallocFileProgress(Percent As Long, Position As Double, Duration As Double)

#### Description

This [TOnProgress](#) event occurs periodically during the creation of the preallocated file used for recording.

#### See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Output with HelpNDoc's Advanced Project Analyzer](#)

## OnCreatePreallocFileStarted

### TVideoGrabber.OnCreatePreallocFileStarted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the preallocated recording file begins to be created.

#### Declaration

**property** OnCreatePreallocFileStarted: TOnFileNotification **read** FOnCreatePreallocFileStarted **write** FOnCreatePreallocFileStarted;

\_\_property TOnFileNotification OnCreatePreallocFileStarted=read=FOnCreatePreallocFileStarted, write=FOnCreatePreallocFileStarted

Event OnCreatePreallocFileStarted(FileName As String)

#### Description

This [TOnFileNotification](#) event is used when [PreallocCapFileEnabled](#) is true (a preallocated huge capture file is used for recording).

Occurs when the preallocated recording file begins to be created.

#### See Also

[TOnCreatePreallocatedFileCompleted](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileProgress](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## OnDbIClick

### TVideoGrabber.OnDbIClick

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

[ber](#)

normal OnDbClick event

**Declaration**

**property** OnDbClick: TEventNotification **read** FOnDbClick **write** FOnDbClick;

\_\_property TEventNotification OnDbClick=read=FOnDbClick, write=FOnDbClick

Event OnDbClick()

**Description**

normal OnDbClick event

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

**OnDeviceArrivalOrRemoval****TVideoGrabber.OnDeviceArrivalOrRemoval**[Prev](#)[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs each time a video capture device is plugged-in or removed.

**Declaration**

**property** OnDeviceArrivalOrRemoval: TOnDeviceArrivalOrRemoval **read** FOnDeviceArrivalOrRemoval **write** FOnDeviceArrivalOrRemoval;

\_\_property TOnDeviceArrivalOrRemoval OnDeviceArrivalOrRemoval=read=FOnDeviceArrivalOrRemoval, write=FOnDeviceArrivalOrRemoval

Event OnDeviceArrivalOrRemoval(IsDeviceArrival As Boolean, IsVideoDevice As Boolean, DeviceName As String, DeviceIndex As Long)

**Description**

This [TOnDeviceArrivalOrRemoval](#) event occurs each time a video capture device is plugged-in or removed and returns information about this device: audio or video device, connected or disconnected, device name.

The [VideoDevices](#) list or [AudioDevices](#) list is updated if a new capture device has been plugged-in.

If a video capture device has been unplugged, it remains in the list. To determine if a device is connected or not, test the IsVideoDeviceConnected (DeviceIndex: LongInt): Boolean function that will return false after the device removal.

When the video capture device is plugged-in again, IsVideoDeviceConnected (DeviceIndex: LongInt): Boolean returns true again.

(Use IsAudioDeviceConnected to test audio devices from the [AudioDevices](#) list)

Note: device arrival/removal detection works only with WDM drivers, "old" VFW capture device appear always in the device lists, even if not connected.

**See Also**

[TOnDeviceArrivalOrRemoval](#) [OnDeviceLost](#) [RefreshDevicesAndCompressorsLists](#)

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

**OnDeviceLost****TVideoGrabber.OnDeviceLost**[Prev](#)[Next](#)

## [TVideoGrabber](#) [Events](#)

Occurs when a capture device is disconnected during preview or recording.

### Declaration

**property** OnDeviceLost: TEventNotification **read** FOnDeviceLost **write** FOnDeviceLost;

**\_\_property** TEventNotification OnDeviceLost=read=FOnDeviceLost, write=FOnDeviceLost

Event OnDeviceLost()

### Description

Occurs when a capture device is disconnected during preview or recording.

This event occurs for IP cameras when the connection is lost and the auto-reconnection is disabled (see [Auto-reconnection](#))

### See Also

[TOnDeviceArrivalOrRemoval](#) [OnDeviceArrivalOrRemoval](#) [RefreshDevicesAndCompressorsLists](#)

Created with the Standard Edition of HelpNDoc: [Eliminate the Struggles of Documentation with a Help Authoring Tool](#)

## OnDeviceReconnected

### TVideoGrabber.OnDeviceReconnected

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Events](#)

Occurs when an IP camera is reconnected after the connection has been lost

### Declaration

**property** OnDeviceReconnected: TEventNotification **read** FOnDeviceReconnected **write** FOnDeviceReconnected;

**\_\_property** TEventNotification OnDeviceReconnected=read=FOnDeviceReconnected, write=FOnDeviceReconnected

Event OnDeviceReconnected()

### Description

Occurs when an IP camera is reconnected, after the connection has been lost

See [Auto-reconnection](#)

Created with the Standard Edition of HelpNDoc: [Free Qt Help documentation generator](#)

## OnDeviceReconnecting

### TVideoGrabber.OnDeviceReconnecting

[Prev](#)

[Next](#)

#### [TVideoGrabber](#) [Events](#)

Occurs when the connection to an IP camera has been lost and the RTSP filter tries to reconnect

### Declaration

**property** OnDeviceReconnecting: TEventNotification **read** FOnDeviceReconnecting **write** FOnDeviceReconnecting;

**\_\_property** TEventNotification OnDeviceReconnecting=read=FOnDeviceReconnecting,



write=FOnDeviceReconnecting

Event OnDeviceReconnecting()

### Description

Occurs when the connection to an IP camera has been lost and the RTSP filter tries to reconnect  
See [Auto-reconnection](#)

Created with the Standard Edition of HelpNDoc: [Converting Word Documents to eBooks: A Step-by-Step Guide with HelpNDoc](#)

## OnDirectNetworkStreamingHostUrl

### TVideoGrabber.OnDirectNetworkStreamingHostUrl

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Returns the host url to use for direct network streaming.

### Declaration

**property** OnDirectNetworkStreamingHostUrl: TOnDirectNetworkStreamingHostUrl **read** FOnDirectNetworkStreamingHostUrl **write** FOnDirectNetworkStreamingHostUrl;

\_\_property TOnDirectNetworkStreamingHostUrl  
OnDirectNetworkStreamingHostUrl=read=FOnDirectNetworkStreamingHostUrl,  
write=FOnDirectNetworkStreamingHostUrl

Event OnDirectNetworkStreamingHostUrl(HostUrl As String, HostName As String, HostPort As Long)

### Description

This [TOnDeviceArrivalOrRemoval](#) event is used to retrieve the host url when performing direct streaming.  
Each user must connect to this URL on the platform where TVideoGrabber is streaming.  
Returns the full URL, the host name and the port.

### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#)  
[ASFVideoBitRate](#) [ASFVideoChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#)  
[ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#)  
[ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#)  
[ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#)  
[ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#)  
[NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [ShowDialog](#)  
[StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## OnDiskFull

### TVideoGrabber.OnDiskFull

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the disk becomes full during recording.

### Declaration

**property** OnDiskFull: TEventNotification **read** FOnDiskFull **write** FOnDiskFull;

\_\_property TEventNotification OnDiskFull=read=FOnDiskFull, write=FOnDiskFull

Event OnDiskFull()

## Description

Occurs when the disk where is located the [StoragePath](#) becomes full during recording, which is stopped. This event occurs also when the space remaining on disk is lower than 25 Mb.

## See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

## OnDragDropFiles

### TVideoGrabber.OnDragDropFiles

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when dropping file(s)

## Declaration

**property** OnDragDropFiles: TOnDragDropFiles **read** FOnDragDropFiles **write** FOnDragDropFiles;

\_\_**property** TOnDragDropFiles OnDragDropFiles=read=FOnDragDropFiles, write=FOnDragDropFiles

Event OnDragDropFiles

## Description

Occurs when dropping file(s) over the TVideoGrabber control (for Delphi / C++Builder)

Type [TOnDragDropFiles](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## OnDVCommandCompleted

### TVideoGrabber.OnDVCommandCompleted

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a DV VCR command is completed.

## Declaration

**property** OnDVCommandCompleted: TOnDVCommandCompleted **read** FOnDVCommandCompleted **write** FOnDVCommandCompleted;

\_\_**property** TOnDVCommandCompleted OnDVCommandCompleted=read=FOnDVCommandCompleted, write=FOnDVCommandCompleted

Event OnDVCommandCompleted(NewStateValue As Long, NewStateLabel As String)

## Description

This [TOnDVCommandCompleted](#) event occurs:

- occurs also when invoking [StartPreview](#) or [StartRecording](#), to retrieve the current state of the DV device.
- occurs after invoking [SendDVCommand](#), when a DV VCR command is completed.
- occurs when the DV VCR state changes (e.g. when rewind ends).

## See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRRecordingInNativeFormatSeparatesStreams](#) [DVRReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Don't be left in the past: convert your WinHelp HLP help files to CHM with HelpNDoc](#)

## OnDVDDiscontinuity

### TVideoGrabber.OnDVDDiscontinuity

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a DV a discontinuity occurs

## Declaration

**property** OnDVDDiscontinuity: TOnDVDDiscontinuity **read** FOnDVDDiscontinuity **write** FOnDVDDiscontinuity;

\_\_property TOnDVDDiscontinuity OnDVDDiscontinuity=read=FOnDVDDiscontinuity, write=FOnDVDDiscontinuity

Event OnDVDDiscontinuity( DeliverNewFrame as Boolean)

## Description

Occurs when a time discontinuity is detected in the DV date/time, if available.

From this event you can invoke e.g. [StopRecording](#) to stop the current recording, or [RecordToNewFileNow](#) to start a new recording when a time discontinuity is detected in the DV date/time data.

Set the **DeliverNewFrame** parameter to "false" to prevent the current video frame (that corresponds to the 1st frame of the new time) to be saved in the file generated when closing the current recording.

**Important: from this event you should not perform any actions with the potential to block, such as holding a critical section or waiting on another thread. Also, do not call any GDI or USER32.DLL APIs.**

Created with the Standard Edition of HelpNDoc: [Transform Your Word Document into a Professional eBook with HelpNDoc](#)

## OnEnumerateWindows

### TVideoGrabber.OnEnumerateWindows

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Return information about the visible windows.

## Declaration

**property** OnEnumerateWindows: T OnEnumerateWindows **read** FOnEnumerateWindows **write** FOnEnumerateWindows;

\_\_property TOnEnumerateWindows OnEnumerateWindows==FOnEnumerateWindows,

write=FOnEnumerateWindows;

Event OnEnumerateWindow (WindowHandle as Long, WindowName as String, WindowClass as string)

### Description

Can be used in the screen recording of a window, to retrieve the name, class and handle of the visible windows.

This lets you retrieve the identifier of a window to perform the screen recording on this window.

See the "[Recording a window](#)" chapter for more information.

Created with the Standard Edition of HelpNDoc: [Step-by-Step Guide: How to Turn Your Word Document into an eBook](#)

## OnFilterSelected

### TVideoGrabber.OnFilterSelected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a filter is selected.

### Declaration

**property** OnFilterSelected: TOnFilterSelected **read** FOnFilterSelected **write** FOnFilterSelected;

**\_\_property** TOnFilterSelected OnFilterSelected==FOnFilterSelected, write=FOnFilterSelected;

Event OnFilterSelected (FilterName as String, ByRef RejectFilter as Boolean)

### Description

This [TOnFilterSelected](#) event occurs when filters are selected automatically when constructing the graph. The FilterName parameter lets you know what is the filter selected. The RejectFilter parameter lets you reject the filter.

Created with the Standard Edition of HelpNDoc: [Effortlessly optimize your documentation website for search engines](#)

## OnFrameBitmap

### TVideoGrabber.OnFrameBitmap

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Gives access to each frame bitmap

### Declaration

**property** OnFrameBitmap: TOnFrameBitmap **read** FOnFrameBitmap **write** FOnFrameBitmap;

**\_\_property** TOnFrameBitmap OnFrameBitmap=read=FOnFrameBitmap, write=FOnFrameBitmap

Event OnFrameBitmap(ByVal FrameInfo As Long, ByVal BitmapInfo As Long)

### Description

This event occurs for each frame bitmap when the [FrameGrabber](#) is enabled.

To preserve the performance while offering a lot of useful values, this event returns 2 parameters that are pointers to structures:

A) the **FrameInfo** parameter, that is a pointer to a [TFrameInfo](#) structure.  
It returns time and frame number information about the current video frames

B) the **BitmapInfo** parameter, which is a pointer to a [TFrameBitmapInfo](#) structure.  
It returns all the useful bitmap information, and also the current mouse information (current position, last clicked position, button states, mouse down, etc...)

From this event you can perform powerful bitmap tasks, like line drawing, mouse-based drawing, pixel modifications, etc...

*You will find a lot of sample code in the OnFrameBitmap event of the MainDemo project (search "FrameBitmap" in the MainForm)*

*To activate this sample code, run the MainDemo project -> the checkboxes are located in the "Overlay" tab.*

NOTE:

**1. from this event you should not perform any actions with the potential to block:**

1. do not hold a critical section or wait on another thread,
2. do not call any GDI or USER32.DLL APIs that might cause a window to move,
3. do not invoke component properties that send messages, e.g. do not read the ItemIndex property of a ListBox component.

If you need to read a such property, set an intermediary variable when the property changes, and then read the intermediary variable from the OnFrameOverlay... event.

**2. if you need to perform blocking operations from this event, enable the [OnFrameBitmapEventSynchronone](#) property.**

*Enabling the OnFrameBitmapEventSynchronone property may slow down the application and/or introduce latency in the video stream.*

**See Also**

[TCardinalDirection](#) [TAutoFileName](#) [TFrameCaptureDest](#) [TFrameGrabberRGBFormat](#)  
[TOnFrameCaptureCompleted](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[TVideoDeinterlacing](#) [AdjustOverlayAspectRatio](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#)  
[CaptureFrameSyncTo](#) [CaptureFrameTo](#) [DrawBitmapOverFrame](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#)  
[FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [FrameGrabber](#) [FrameGrabberCurrentRGBFormat](#)  
[FrameGrabberRGBFormat](#) [FramerateDivider](#) [GetFrameInfo](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#)  
[GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#)  
[ImageOverlay](#) [StretchToVideoSize](#) [InFrameProgressEvent](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#)  
[JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#)  
[MouseWheelEventEnabled](#) [MP4NeedsReindexing](#) [OnDiskFull](#) [OnFrameBitmapEventSynchronone](#)  
[OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#)  
[OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#)  
[RetrievalInitialXYAfterRotation](#) [SetFrameCaptureBounds](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [StoragePath](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)

[TranslateMouseCoordinates](#) [VideoProcessing](#) [Brightness](#) [VideoProcessing](#) [Contrast](#)  
[VideoProcessing](#) [Deinterlacing](#) [VideoProcessing](#) [FlipHorizontal](#) [VideoProcessing](#) [FlipVertical](#)  
[VideoProcessing](#) [GrayScale](#) [VideoProcessing](#) [Hue](#) [VideoProcessing](#) [InvertColors](#)  
[VideoProcessing](#) [Pixellization](#) [VideoProcessing](#) [Rotation](#) [VideoProcessing](#) [RotationCustomAngle](#)  
[VideoProcessing](#) [Saturation](#) [WebcamStillCaptureButton](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: Streamline Your Documentation Process with HelpNDoc's Intuitive Interface

## OnFrameCaptureCompleted

### TVideoGrabber.OnFrameCaptureCompleted

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Events](#)  
[ber](#)

Returns each video frame captured.

#### Declaration

**property** OnFrameCaptureCompleted: TOnFrameCaptureCompleted **read** FOnFrameCaptureCompleted **write** FOnFrameCaptureCompleted;

**\_\_property** TOnFrameCaptureCompleted

OnFrameCaptureCompleted=read=FOnFrameCaptureCompleted, write=FOnFrameCaptureCompleted

Event OnFrameCaptureCompleted(FrameBitmapHandle As Long, BitmapWidth As Long, BitmapHeight As Long, FrameNumber As Long, FrameTime As Double, DestType As TxFramCaptureDest, FileName As String, Success As Boolean, Frameld As Long)

#### Description

This [TOnFrameCaptureCompleted](#) event returns each video frame captured by using [CaptureFrameTo](#) or the [burst mode](#) .

For Delphi and C++Buidler versions, the video frame is returned as TBitmap (from the "graphics" unit). For OCX versions the video frame is returned as a Bitmap handle.

For frame capture to BMP or JPEG files, the file name generated automatically is also returned by this event.

You can get various frame infos from this event by using [GetFrameInfo](#) (FrameId, ...).

Note: the [frame grabber](#) must be enabled to capture video frames.

Sample code is included in the MainDemo project.

To get information about the current video frame, invoke [GetFrameInfo](#) (FrameId, ...). E.g.

```
SampleTimeHour = VideoGrabber.GetFrameInfo (FrameId, fi_SampleTime_Hour)
```

#### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#)  
[BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#)  
[FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#)  
[GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#)  
[JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#)  
[Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#)  
[WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Free iPhone documentation generator](#)



## OnFrameOverlayUsingDC

### TVideoGrabber.OnFrameOverlayUsingDC

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Events](#)

Put your code in this event to draw graphic objects over video frames by using the frame DC.

#### Declaration

**property** OnFrameOverlayUsingDC: TOnFrameOverlayUsingDC **read** FOnFrameOverlayUsingDC **write** SetOnFrameOverlayUsingDC;

\_\_property TOnFrameOverlayUsingDC OnFrameOverlayUsingDC=read=FOnFrameOverlayUsingDC, write=SetOnFrameOverlayUsingDC

Event OnFrameOverlayUsingDC(Dc As Long, FrameNumber As Long, FrameTime As Double, FrameId As Long)

#### Description

**DEPRECATED, use [OnFrameBitmap](#) instead**

This [TOnFrameOverlayUsingDC](#) event is used to draw graphic objects over video frames, by using the frame device context.

This event occurs for each video frame that goes through the video stream, during preview, recording or playback.

The [frame grabber](#) must be enabled for this event to occur.

From this event it is possible to:

- draw a bitmap over frames by calling DrawBitmapOverFrame,
- draw anything using the Dc parameter (device context) and GDI functions.

This event is TIME CRITICAL. The code used in this event must be as short and fast as possible.

Sample code is included in the MainDemo project.

To get information about the current video frame, invoke [GetFrameInfo](#) (FrameId, ...). E.g.

```
SampleTimeHour = VideoGrabber.GetFrameInfo (FrameId, fi_SampleTime_Hour)
```

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)



[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Keep Your PDFs Safe from Unauthorized Access with These Security Measures](#)

## OnFrameOverlayUsingDIB

### TVideoGrabber.OnFrameOverlayUsingDIB

[Prev](#)
[Next](#)

[TVideoGrabber](#)  
[Events](#)

Put your code in this event to apply any processing to each video frame, through its DIB bitmap.

#### Declaration

**property** OnFrameOverlayUsingDIB: TOnFrameOverlayUsingDIB **read** FOnFrameOverlayUsingDIB **write** FOnFrameOverlayUsingDIB;

\_\_property TOnFrameOverlayUsingDIB OnFrameOverlayUsingDIB=read=FOnFrameOverlayUsingDIB, write=FOnFrameOverlayUsingDIB

Event OnFrameOverlayUsingDIB(FrameBitmapHandle As Long, FrameNumber As Long, FrameTime As Double, FrameId As Long)

#### Description

**DEPRECATED, use [OnFrameBitmap](#) instead**

This TOnFrameOverlayUsingDIB event is used to process each video frame DIB bitmap.

This event occurs for each video frame that goes through the video stream, during preview, recording or playback.

The [frame grabber](#) must be enabled for this event to occur.

This event offers a full access to the DIB bitmap of each video frame.

This event is TIME CRITICAL. The code used in this event must be as short and fast as possible.

Sample code is included in the MainDemo project.

To get information about the current video frame, invoke [GetFrameInfo](#) (FrameId, ...). E.g.

```
SampleTimeHour = VideoGrabber.GetFrameInfo (FrameId, fi_SampleTime_Hour)
```

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay\\_AlphaBlendValue](#) [SetImageOverlay\\_ChromaKey](#)  
[SetImageOverlay\\_ChromaKeyLeewayPercent](#) [SetImageOverlay\\_ChromaKeyRGBColor](#)  
[SetImageOverlay\\_Enabled](#) [SetImageOverlay\\_Height](#) [SetImageOverlay\\_LeftLocation](#)

[SetImageOverlay\\_RotationAngle](#) [SetImageOverlay\\_StretchToVideoSize](#) [SetImageOverlay\\_TargetDisplay](#)  
[SetImageOverlay\\_TopLocation](#) [SetImageOverlay\\_Transparent](#) [SetImageOverlay\\_TransparentColorValue](#)  
[SetImageOverlay\\_UseTransparentColor](#) [SetImageOverlay\\_Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay\\_Align](#) [SetTextOverlay\\_BkColor](#) [SetTextOverlay\\_CustomVar](#) [SetTextOverlay\\_Enabled](#)  
[SetTextOverlay\\_Font](#) [SetTextOverlay\\_FontColor](#) [SetTextOverlay\\_GradientColor](#)  
[SetTextOverlay\\_GradientMode](#) [SetTextOverlay\\_HighResFont](#) [SetTextOverlay\\_Left](#) [SetTextOverlay\\_Right](#)  
[SetTextOverlay\\_Scrolling](#) [SetTextOverlay\\_ScrollingSpeed](#) [SetTextOverlay\\_Shadow](#)  
[SetTextOverlay\\_ShadowColor](#) [SetTextOverlay\\_ShadowDirection](#) [SetTextOverlay\\_String](#)  
[SetTextOverlay\\_TargetDisplay](#) [SetTextOverlay\\_Top](#) [SetTextOverlay\\_Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay\\_Align](#) [TextOverlay\\_BkColor](#)  
[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#)  
[TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#)  
[TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#)  
[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Free Web Help generator](#)

## OnFrameOverlayUsingVIDEOHDR

### TVideoGrabber.OnFrameOverlayUsingVIDEOHDR

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Put your code in this event to apply any processing to each video frame, through a VFW VIDEOHDR format.

#### Declaration

**property** OnFrameOverlayUsingVIDEOHDR: TOnFrameOverlayUsingVIDEOHDR **read** FOnFrameOverlayUsingVIDEOHDR **write** FOnFrameOverlayUsingVIDEOHDR;

\_\_property TOnFrameOverlayUsingVIDEOHDR  
 OnFrameOverlayUsingVIDEOHDR=read=FOnFrameOverlayUsingVIDEOHDR,  
 write=FOnFrameOverlayUsingVIDEOHDR

n/a

#### Description

**DEPRECATED, use [OnFrameBitmap](#) instead**

This TOnFrameOverlayUsingVIDEOHDR event is used to draw graphic objects over video frames, by using a VIDEOHDR video frame format.

This event occurs for each video frame that goes through the video stream, during preview, recording or playback.

The [frame grabber](#) must be enabled for this event to occur.

Sample code is included in the MainDemo project.

To get information about the current video frame, invoke [GetFrameInfo](#) (FrameId, ...). E.g.

```
SampleTimeHour = VideoGrabber.GetFrameInfo (FrameId, fi_SampleTime_Hour)
```

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay\\_StretchToVideoSize](#)

[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## OnFrameProgress

### TVideoGrabber.OnFrameProgress

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Returns information about each video frame.

#### Declaration

**property** OnFrameProgress: TOnFrameProgress **read** FOnFrameProgress **write** FOnFrameProgress;

\_\_property TOnFrameProgress OnFrameProgress=read=FOnFrameProgress, write=FOnFrameProgress

Event OnFrameProgress(FrameNumber As Long, FrameTime As Double, FrameId As Long)

#### Description

**DEPRECATED, use [OnFrameProgress2](#) instead**

This [TOnFrameProgress](#) event occurs asynchronously for each video frame.

To get information about the current video frame within this event, invoke [GetFrameInfo](#) (FrameId, ...).  
E.g.

```
SampleTimeHour = VideoGrabber.GetFrameInfo (FrameId, fi_SampleTime_Hour)
```

*Look at the [VideoGrabberFrameProgress](#) event in the [MainDemo](#) project for sample code.*

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#)  
[Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#)  
[IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#)  
[PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#)

[UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Make Documentation Review a Breeze with HelpNDoc's Advanced Project Analyzer](#)

## OnFrameProgress2

### TVideoGrabber.OnFrameProgress2

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs for each video frame

#### Declaration

**property** OnFrameProgress2: TOnFrameProgress2 **read** FOnFrameProgress2 **write** FOnFrameProgress2;

\_\_property TOnFrameProgress2 OnFrameProgress2 =read=FOnFrameProgress2, write=FOnFrameProgress2

Event OnFrameProgress2 (ByVal FrameInfo As Long)

#### Description

This event occurs asynchronously for each video frame.

It returns the current frame time, frame number, DV frame time, timecode, etc... through the FrameInfo pointer to the [TFrameInfo](#) structure.

*You will find sample code in the MainForm of the MainDemo project ( Search for "FrameProgress2")*

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## OnGraphBuilt

### TVideoGrabber.OnGraphBuilt

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the video graph is built.

#### Declaration

**property** OnGraphBuilt: TNotifyEvent **read** FOnGraphBuilt **write** FOnGraphBuilt;

\_\_property Classes::TNotifyEvent OnGraphBuilt=read=FOnGraphBuilt, write=FOnGraphBuilt

Event OnGraphBuilt()

#### Description

Occurs when a preview, recording or playback video graph is built.

Used mainly to retrieve properties that are available only after the video graph is built, like [GetVMR9ImageAdjustmentBounds](#).

E.g.:

```
procedure TForm1.VideoGrabberGraphBuilt(Sender: TObject);
var
    MinValue, MaxValue, StepSize, DefaultValue, CurrentValue: LongInt;
begin
```

```

    BrightnessTrackbar.Enabled := VideoGrabber.GetVMR9ImageAdjustmentBounds (True, vmr9_Brightness, tbrVMR9Brightness.Position);
    if Brightness.Enabled then begin
        Brightness.Min := MinValue;
        Brightness.Max := MaxValue;
        Brightness.Frequency := StepSize;
        Brightness.Position := CurrentValue;
    end;
end;

procedure TForm1.BrightnessTrackbarChange(Sender: TObject);
begin
    VideoGrabber.SetVMR9ImageAdjustmentValue (True, vmr9_Brightness, tbrVMR9Brightness.Position);
end;

```

See the MainDemo project for sample code.

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## OnInactive

### TVideoGrabber.OnInactive

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the current task ends.

#### Declaration

**property** OnInactive: TEventNotification **read** FOnInactive **write** FOnInactive;

\_\_property TEventNotification OnInactive=read=FOnInactive, write=FOnInactive

Event OnInactive()

#### Description

Occurs when the current task ends, after invoking [StopPreview](#), [StopRecording](#), [ClosePlayer](#), etc...

Created with the Standard Edition of HelpNDoc: [Free EBook and documentation generator](#)

## OnKeyPress

### TVideoGrabber.OnKeyPress

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when keys are pressed on a video window that has the control.

#### Declaration

**property** OnKeyPress: TOnVideoKeyPress **read** FOnKeyPress **write** FOnKeyPress;

\_\_property TOnVideoKeyPress OnKeyPress=read=FOnKeyPress, write=FOnKeyPress

Event OnKeyPress(VideoWindow As Long, Key, PhysicalKey As Long)

#### Description

This [TOnVideoKeyPress](#) event occurs when keys are pressed on a video window that has the control.

Useful mainly when the video window is not attached to the control ([Display embedded](#) disabled), to receive key events from the window.

## See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Free Qt Help documentation generator](#)

## OnLastCommandCompleted

### TVideoGrabber.OnLastCommandCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a command has been completed

#### Declaration

**property** OnLastCommandCompleted: TOnEventNotification **read** FOnLastCommandCompleted **write** FOnLastCommandCompleted;

**\_\_property** TOnEventNotification OnLastCommandCompleted=read=FOnLastCommandCompleted, write=FOnLastCommandCompleted

Event OnLastCommandCompleted

#### Description

Occurs when a command has been completed (e.g. StartPreview, OpenPlayer, StopRecording, etc...)

Created with the Standard Edition of HelpNDoc: [Free Qt Help documentation generator](#)

## OnLeavingFullScreen

### TVideoGrabber.OnLeavingFullScreen

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when leaving the full screen mode

#### Declaration

**property** OnLeavingFullScreen: TOnEventNotification **read** FOnLeavingFullScreen **write** FOnLeavingFullScreen;

**\_\_property** TOnEventNotification OnLeavingFullScreen==FOnLeavingFullScreen, write=FOnLeavingFullScreen;

Event OnLeavingFullScreen()

#### Description

This event occurs when leaving the full screen mode, e.g. when the user press the "ESC" key

## See Also

[TVideoRenderer](#) [AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#)

[DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#)  
[DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#)  
[DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#)  
[IsVideoPortAvailable](#) [Monitor\\_Primary\\_Index](#) [MonitorBounds](#) [MonitorsCount](#) [SetParentWindow](#)  
[SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#)  
[VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#)  
[VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

## OnLog

### TVideoGrabber.OnLog

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Returns information, warning and error messages

#### Declaration

**property** OnLog: TOnLog **read** FOnLog **write** FOnLog;

\_\_property TOnLog OnLog=read=FOnLog, write=FOnLog

Event OnLog(LogType As TxLogType, Severity As String, InfoMsg As String)

#### Description

Warnings and error messages are returned by this event.

See also [TOnLog](#) and [GetLogString](#)

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

## OnMotionDetected

### TVideoGrabber.OnMotionDetected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a new frame is received and motion is detected.

#### Declaration

**property** OnMotionDetected: TOnMotionDetected **read** FOnMotionDetected **write** FOnMotionDetected;

\_\_property TOnMotionDetected OnMotionDetected=read=FOnMotionDetected, write=FOnMotionDetected

Event OnMotionDetected(GlobalMotionRatio As Double, MaxMotionCellX As Long, MaxMotionCellY As Long, FrameBitmapHandle As Long, FrameNumber As Long, FrameTime As Double, FrameId As Long, CaptureFrame)

#### Description

This [TOnMotionDetected](#) event occurs when a new frame is received and motion is detected.

Note: be sure to activate the motion detector by enabling [MotionDetector\\_Enabled](#).

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#)  
[Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#)



[MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#)  
[OnMotionNotDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer](#)

## OnMotionNotDetected

### TVideoGrabber.OnMotionNotDetected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a frame is received and no motion is detected.

#### Declaration

**property** OnMotionNotDetected: TOnMotionNotDetected **read** FOnMotionNotDetected **write** FOnMotionNotDetected;

\_\_property TOnMotionNotDetected OnMotionNotDetected=read=FOnMotionNotDetected, write=FOnMotionNotDetected

Event OnMotionNotDetected(FrameBitmapHandle As Long, FrameNumber As Long, FrameTime As Double, FrameId As Long, CaptureFrame)

#### Description

This [TOnMotionNotDetected](#) event occurs when a frame is received and no motion is detected.

Note: be sure to activate the motion detector by enabling [MotionDetector\\_Enabled](#).

#### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#)  
[Video noise](#) [TOnMotionDetected](#) [TOnMotionNotDetected](#) [MotionDetector\\_CellMotionRatio](#)  
[MotionDetector\\_CompareBlue](#) [MotionDetector\\_CompareGreen](#) [MotionDetector\\_CompareRed](#)  
[MotionDetector\\_Enabled](#) [MotionDetector\\_EnumGridDialogControls](#) [MotionDetector\\_Get2DTextGrid](#)  
[MotionDetector\\_Get2DTextMotion](#) [MotionDetector\\_GetCellLocation](#) [MotionDetector\\_GetCellSensitivity](#)  
[MotionDetector\\_GetCellSize](#) [MotionDetector\\_GloballyIncOrDecSensitivity](#)  
[MotionDetector\\_GlobalMotionRatio](#) [MotionDetector\\_GreyScale](#) [MotionDetector\\_Grid](#)  
[MotionDetector\\_GridXCount](#) [MotionDetector\\_GridYCount](#) [MotionDetector\\_IsGridValid](#)  
[MotionDetector\\_MaxDetectionsPerSecond](#) [MotionDetector\\_ReduceCPULoad](#)  
[MotionDetector\\_ReduceVideoNoise](#) [MotionDetector\\_Reset](#) [MotionDetector\\_ResetGlobalSensitivity](#)  
[MotionDetector\\_SetCellSensitivity](#) [MotionDetector\\_SetGridSize](#) [MotionDetector\\_ShowGridDialog](#)  
[MotionDetector\\_Triggered](#) [MotionDetector\\_UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#)  
[OnMotionDetected](#) [RecordingOnMotion\\_Enabled](#) [RecordingOnMotion\\_MotionThreshold](#)  
[RecordingOnMotion\\_NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Encrypted, Password-Protected PDFs](#)

## OnMouseDown

### TVideoGrabber.OnMouseDown

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user presses a mouse button with the mouse pointer over a control.

#### Declaration

**property** OnMouseDown: TOnVideoMouseUpDown **read** FOnMouseDown **write** FOnMouseDown;

\_\_property TOnVideoMouseUpDown OnMouseDown=read=FOnMouseDown, write=FOnMouseDown

Event OnMouseDown(VideoWindow As Long, Button As TMouseButton, x As Long, y As Long)

### Description

- if [TranslateMouseCoordinates](#) is disabled, the normal OnMouseUp event occurs and returns the real mouse coordinates across the control.
- if [TranslateMouseCoordinates](#) is enabled, this event returns the coordinates corresponding to the native video size.

The difference appears when [Display\\_AutoSize](#) is disabled or [PreviewZoomSize](#) <> 100. In this case, the event reports the corresponding native video source coordinate, necessary to draw objects over video frames from the [OnFrameBitmap](#) event.

*Note that the BitmapInfo parameter of the [OnFrameBitmap](#) event returns directly the current mouse information.*

**Note:** to detect if the video is displayed when the event occurs, test VideoWindow <> -1.

E.g.:

```
procedure TfrmMainForm.VideoGrabberMouseDown(Sender: TObject;
  VideoWindow: Integer; Button: TMouseButton; Shift: TShiftState; X,
  Y: Integer);
begin
  if VideoWindow <> -1 then begin // if the video is displayed
    if chkFreeHandEnabled.Checked then begin
      ...
      ...
    end;
  end;
end;
```

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)

## [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## OnMouseDown\_Video

### TVideoGrabber.OnMouseDown\_Video

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user presses a mouse button with the mouse pointer over a control.

#### Declaration

**property** OnMouseDown\_Video: TOnVideoMouseUpDown **read** FOnMouseDown **write** FOnMouseDown;

\_\_property TOnVideoMouseUpDown OnMouseDown\_Video=read=FOnMouseDown, write=FOnMouseDown

Event OnMouseDown\_Video(VideoWindow As Long, Button As TxMouseButton, x As Long, y As Long)

#### Description

returns the coordinates of the original video frame at the cursor location

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

## OnMouseDown\_Window

### TVideoGrabber.OnMouseDown\_Window

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user presses a mouse button with the mouse pointer over a control.

#### Declaration

**property** OnMouseDown\_Window: TOnVideoMouseUpDown **read** FOnMouseDown **write** FOnMouseDown;

\_\_property TOnVideoMouseUpDown OnMouseDown\_Window=read=FOnMouseDown, write=FOnMouseDown

Event OnMouseDown\_Window(VideoWindow As Long, Button As TxMouseButton, x As Long, y As Long)

#### Description

returns the coordinates of the video window at the cursor location

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

## OnMouseMove

### TVideoGrabber.OnMouseMove

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user moves the mouse pointer while the mouse pointer is over a control.

#### Declaration

**property** OnMouseMove: TOnVideoMouseMove **read** FOnMouseMove **write** FOnMouseMove;

\_\_property TOnVideoMouseMove OnMouseMove=read=FOnMouseMove, write=FOnMouseMove

Event OnMouseMove(VideoWindow As Long, x As Long, y As Long)

### Description

**- if [TranslateMouseCoordinates](#) is enabled:**

this event returns the video frame coordinates corresponding to the native video size. **Therefore the event does not occur if the mouse is moved out of the video window.**

**- if [TranslateMouseCoordinates](#) is disabled:**

the normal OnMouseUp event occurs and returns the usual mouse coordinates **over the video window, as well as out of the video window** if the mouse has been clicked within the video window and then moved out of it.

The difference appears when [Display AutoSize](#) is disabled or [PreviewZoomSize](#) <> 100. In this case, the event reports the corresponding native video source coordinate, necessary to draw objects over video frames from the [OnFrameBitmap](#) event.

*Note that the BitmapInfo parameter of the [OnFrameBitmap](#) event returns directly the current mouse information.*

**Note:** to detect if the video is displayed when the event occurs, test VideoWindow <> -1.

E.g.:

```
procedure TfrmMainForm.VideoGrabberMouseMove(Sender: TObject;
  VideoWindow: Integer; Shift: TShiftState; X, Y: Integer);
begin
  if VideoWindow <> -1 then begin // if the video is displayed
    if chkFreeHandEnabled.Checked then begin
      ...
      ...
    end;
  end;
end;
```

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#)  
[AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#)  
[MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#)  
[OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)

[TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's User-Friendly UI](#)

## OnMouseMove\_Video

### TVideoGrabber.OnMouseMove\_Video

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user moves the mouse pointer while the mouse pointer is over a control.

#### Declaration

**property** OnMouseMove\_Video: TOnVideoMouseMove **read** FOnMouseMove **write** FOnMouseMove;

\_\_property TOnVideoMouseMove OnMouseMove\_Video=read=FOnMouseMove, write=FOnMouseMove

Event OnMouseMove\_Video(VideoWindow As Long, x As Long, y As Long)

#### Description

returns the coordinates of the original video frame at the cursor location

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## OnMouseMove\_Window

### TVideoGrabber.OnMouseMove\_Window

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user moves the mouse pointer while the mouse pointer is over a control.

#### Declaration

**property** OnMouseMove\_Window: TOnVideoMouseMove **read** FOnMouseMove **write** FOnMouseMove;

\_\_property TOnVideoMouseMove OnMouseMove\_Window=read=FOnMouseMove, write=FOnMouseMove

Event OnMouseMove\_Window(VideoWindow As Long, x As Long, y As Long)

#### Description

returns the coordinates of the video window at the cursor location

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## OnMouseUp

### TVideoGrabber.OnMouseUp

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user releases a mouse button that was pressed with the mouse pointer over a component.

#### Declaration

**property** OnMouseUp: TOnVideoMouseUpDown **read** FOnMouseUp **write** FOnMouseUp;

\_\_property TOnVideoMouseUpDown OnMouseUp=read=FOnMouseUp, write=FOnMouseUp

Event OnMouseUp(VideoWindow As Long, Button As TMouseButton, x As Long, y As Long)

### Description

- if [TranslateMouseCoordinates](#) is disabled, the normal OnMouseUp event occurs and returns the real mouse coordinates across the control.
- if [TranslateMouseCoordinates](#) is enabled, this event returns the coordinates corresponding to the native video size.

The difference appears when [Display\\_AutoSize](#) is disabled or [PreviewZoomSize](#) <> 100. In this case, the event reports the corresponding native video source coordinate, necessary to draw objects over video frames from the [OnFrameBitmap](#) event.

*Note that the [BitmapInfo](#) parameter of the [OnFrameBitmap](#) event returns directly the current mouse information.*

**Note:** to detect if the video is displayed when the event occurs, test VideoWindow <> -1.

E.g.:

```
procedure TfrmMainForm.VideoGrabberMouseUp(Sender: TObject;
  VideoWindow: Integer; Button: TMouseButton; Shift: TShiftState; X,
  Y: Integer);
begin
  if VideoWindow <> -1 then begin // if the video is displayed
    if chkFreeHandEnabled.Checked then begin
      ...
      ...
    end;
  end;
end;
```

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseWheel](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)



## OnMouseUp\_Video

### TVideoGrabber.OnMouseUp\_Video

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user releases a mouse button that was pressed with the mouse pointer over a component.

#### Declaration

**property** OnMouseUp\_Video: TOnVideoMouseUpDown **read** FOnMouseUp **write** FOnMouseUp;

\_\_property TOnVideoMouseUpDown OnMouseUp\_Video=read=FOnMouseUp, write=FOnMouseUp

Event OnMouseUp\_Video(VideoWindow As Long, Button As TxMouseButton, x As Long, y As Long)

#### Description

returns the coordinates of the original video frame at the cursor location

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

## OnMouseUp\_Window

### TVideoGrabber.OnMouseUp\_Window

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user releases a mouse button that was pressed with the mouse pointer over a component.

#### Declaration

**property** OnMouseUp\_Window: TOnVideoMouseUpDown **read** FOnMouseUp **write** FOnMouseUp;

\_\_property TOnVideoMouseUpDown OnMouseUp\_Window=read=FOnMouseUp, write=FOnMouseUp

Event OnMouseUp\_Window(VideoWindow As Long, Button As TxMouseButton, x As Long, y As Long)

#### Description

returns the coordinates of the video window at the cursor location

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

## OnMouseWheel

### TVideoGrabber.OnMouseWheel

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the user turns the mouse wheel up or down

#### Declaration

**property** OnMouseWheel: TOnMouseWheel **read** FOnMouseWheel **write** FOnMouseWheel;

\_\_property TOnMouseWheel OnMouseWheel==FOnMouseWheel, write=FOnMouseWheel;

Event OnMouseWheel (VideoWindow As Long, DirectionUp as Bool, x As Long, y As Long)

#### Description

Occurs when the user turns the mouse wheel up or down and the pointer is over the video window.



To activate this event enable the [MouseWheelEventEnabled](#) property (disabled by default)

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#) [DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#) [OnFrameBitmap](#) [OnFrameBitmapEventSynchronone](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#) [OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#) [SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#) [SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#) [SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#) [SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#) [SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#) [SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#) [SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#) [SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#) [SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#) [SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#) [SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#) [SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#) [SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#) [SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#) [SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#) [ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#) [TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#) [TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#) [TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#) [TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: Effortlessly Create Encrypted, Password-Protected PDFs

## OnNoVideoDevices

### TVideoGrabber.OnNoVideoDevices

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs if no video device is detected.

#### Declaration

**property** OnNoVideoDevices: TEventNotification **read** FOnNoVideoDevices **write** FOnNoVideoDevices;

\_\_property TEventNotification OnNoVideoDevices=read=FOnNoVideoDevices, write=FOnNoVideoDevices

Event OnNoVideoDevices()

#### Description

Occurs if no video device is currently installed or plugged-in on the current platform.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnVideoDeviceSelected](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: Experience a User-Friendly Interface with HelpNDoc's Documentation Tool

## OnNTPTimeStamp

### TVideoGrabber.OnNTPTimeStamp

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Events](#)

Reports the NTP timestamps

#### Declaration

TOnNTPTimeStamp=**procedure** (Sender: TObject; last\_rtcp\_ntp\_time: int64; last\_rtcp\_timestamp: DWORD; delta\_rtcp\_ntp\_time: int64; delta\_rtcp\_timestamp: DWORD; UTCDateTime: TDateTime; LocalDateTime: TDateTime) **of object**;

```
typedef void __fastcall (__closure *TOnNTPTimeStamp)(System::TObject* Sender, __int64 last_rtcp_ntp_time, unsigned last_rtcp_timestamp, __int64 delta_rtcp_ntp_time, unsigned delta_rtcp_timestamp);
```

#### Description

Reports the NTP timestamps if the source is a RTSP URL (requires the Dastead RTSP/RTMP/HTTP/ONVIF Source Filter)

Created with the Standard Edition of HelpNDoc: [Upgrade your help files and your workflow with HelpNDoc's WinHelp HLP to CHM conversion](#)

## OnONVIFDiscoveryCompletedNotification

### TVideoGrabber.OnONVIFDiscoveryCompletedNotification

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Events](#)

Occurs for each ONVIF camera discovered and at the end of the discovery processes

#### Declaration

**property** OnONVIFDiscoveryCompleted: TOnONVIFDiscoveryCompletedNotification **read**  
FOnONVIFDiscoveryCompleted **write** FOnONVIFDiscoveryCompleted;

**\_\_property** TOnONVIFDiscoveryCompletedNotification  
OnONVIFDiscoveryCompleted==FOnONVIFDiscoveryCompleted, write=FOnONVIFDiscoveryCompleted;

#### Description

##### DiscoveryCallbackStatus: TDiscoveryCallbackStatus:

whether the event occurs for a new camera discovered, or for the completion of the Multicast or IP range discovery

##### CameraCount: LongInt:

number of cameras currently discovered

#### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Powerful and User-Friendly Help Authoring Tool for Markdown Documents](#)

## OnOpenURLAsyncStatusChanged

### TVideoGrabber.OnOpenURLAsyncStatusChanged

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Events](#)

URL opening status notification

#### Declaration

property OnOpenURLAsyncStatusChanged: TOnOpenURLAsyncStatusChanged read  
FOnOpenURLAsyncStatusChanged write FOnOpenURLAsyncStatusChanged;

```
__property TOnOpenURLAsyncStatusChanged  
OnOpenURLAsyncStatusChanged==FOnOpenURLAsyncStatusChanged,  
write=FOnOpenURLAsyncStatusChanged;
```

#### Description

This [TOnOpenURLAsyncStatusChanged](#) event occurs when status of the URL being opened is changing

Created with the Standard Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

## OnPlayerBufferingData

### TVideoGrabber.OnPlayerBufferingData

[Prev](#)
[Next](#)
[TVideoGrabber](#) [Events](#)

Occurs when opening an URL in the player and the player is buffering data.

#### Declaration

**property** OnPlayerBufferingData: TOnPlayerBufferingData **read** FOnPlayerBufferingData **write**  
FOnPlayerBufferingData;

```
__property TOnPlayerBufferingData OnPlayerBufferingData=read=FOnPlayerBufferingData,  
write=FOnPlayerBufferingData
```

Event OnPlayerBufferingData(StartingToBuffer As Boolean)

#### Description

Occurs when opening an URL in the player.

This [TOnPlayerBufferingData](#) event occurs just before the clip begins to play:

- a first time when the player starts buffering data (**StartingToBuffer** is true),
- a second time when the player ends buffering data (**StartingToBuffer** is true),

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last](#) [Clip](#) [Played](#) [MP4NeedsReindexing](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help](#)

## OnPlayerEndOfPlaylist

### TVideoGrabber.OnPlayerEndOfPlaylist

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs on the end of the playlist.

#### Declaration

**property** OnPlayerEndOfPlaylist: TEventNotification **read** FOnPlayerEndOfPlaylist **write** FOnPlayerEndOfPlaylist;

\_\_property TEventNotification OnPlayerEndOfPlaylist=read=FOnPlayerEndOfPlaylist, write=FOnPlayerEndOfPlaylist;

Event OnPlayerEndOfPlaylist()

#### Description

Occurs when the playlist is active and the end of the playlist is reached. See the "[Using the playlist](#)" chapter for more information about the playlist feature.

#### See Also

[TPlaylist](#) [Video formats](#) [GetPlaylist](#) [IsPlaylistActive](#) [Playlist](#) [PlaylistIndex](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## OnPlayerEndOfStream

### TVideoGrabber.OnPlayerEndOfStream

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when AVI playback ends.

#### Declaration

**property** OnPlayerEndOfStream: TEventNotification **read** FOnPlayerEndOfStream **write** FOnPlayerEndOfStream;

\_\_property TEventNotification OnPlayerEndOfStream=read=FOnPlayerEndOfStream, write=FOnPlayerEndOfStream;

Event OnPlayerEndOfStream()

#### Description

Occurs during AVI playback, when the end of the video stream is reached.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#)

[VideoWidth PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## OnPlayerOpened

### TVideoGrabber.OnPlayerOpened

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when AVI playback starts.

#### Declaration

**property** OnPlayerOpened: TEventNotification **read** FOnPlayerOpened **write** FOnPlayerOpened;

\_\_property TEventNotification OnPlayerOpened=read=FOnPlayerOpened, write=FOnPlayerOpened

Event OnPlayerOpened()

#### Description

Occurs when AVI playback starts, just before the first video frame is rendered.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronon](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Free PDF documentation generator](#)

## OnPlayerStateChanged

### TVideoGrabber.OnPlayerStateChanged

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the player state changes.

#### Declaration

**property** OnPlayerStateChanged: TOnPlayerStateChanged **read** FOnPlayerStateChanged **write** FOnPlayerStateChanged;

\_\_property TOnPlayerStateChanged OnPlayerStateChanged==FOnPlayerStateChanged, write=FOnPlayerStateChanged;

Event OnPlayerStateChanged()

#### Description

This event occurs when the player state changes. It returns the old and the new player [states](#).

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#)

[FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last Clip Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## OnPlayerUpdateTrackbarPosition

### TVideoGrabber.OnPlayerUpdateTrackbarPosition

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the player trackbar position must be updated

#### Declaration

**property** OnPlayerUpdateTrackbarPosition: TOnFrameProgress **read** FOnPlayerUpdateTrackbarPosition **write** FOnPlayerUpdateTrackbarPosition;

\_\_property TOnFrameProgress

OnPlayerUpdateTrackbarPosition=read=FOnPlayerUpdateTrackbarPosition,  
write=FOnPlayerUpdateTrackbarPosition

Event OnPlayerUpdateTrackbarPosition()

#### Description

This event occurs while playback when you must update your trackbar position with the new frame (or time) position reported by this event.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last Clip Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronone](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Reach: Convert Your Word Document to an ePub or Kindle eBook](#)

## OnPreviewStarted

### TVideoGrabber.OnPreviewStarted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when preview starts.

#### Declaration



**property** OnPreviewStarted: TEventNotification **read** FOnPreviewStarted **write** FOnPreviewStarted;  
 \_\_property TEventNotification OnPreviewStarted=read=FOnPreviewStarted, write=FOnPreviewStarted

Event OnPreviewStarted()

### Description

Occurs after calling [StartPreview](#) , just before the first video frame is rendered.

### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

## OnRawAudioSample

### TVideoGrabber.OnRawAudioSample

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Gives access to the current audio sample.

### Declaration

**property** OnRawAudioSample: TOnRawSample **read** FOnRawAudioSample **write** FOnRawAudioSample;

\_\_property TOnRawSample OnRawAudioSample=read=FOnRawAudioSample, write=FOnRawAudioSample

Event OnRawAudioSample(pSampleBuffer As Long, SampleBufferSize As Long, SampleDataLength As Long, FormatType As TxFormatType, pFormat As Long, pBitmapInfoHeader As Long, SampleStartTime As Double, SampleStopTime As Double)

### Description

This [TOnRawSample](#) event gives access to the current audio sample.

### See Also

[TOnRawSample](#) [OnRawVideoSample](#) [RawAudioSampleCapture](#) [RawCaptureAsyncEvent](#) [RawVideoSampleCapture](#)

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

## OnRawVideoSample

### TVideoGrabber.OnRawVideoSample

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Gives access to the current video sample

### Declaration

**property** OnRawVideoSample: TOnRawSample **read** FOnRawVideoSample **write** FOnRawVideoSample;

\_\_property TOnRawSample OnRawVideoSample=read=FOnRawVideoSample, write=FOnRawVideoSample

Event OnRawVideoSample(pSampleBuffer As Long, SampleBufferSize As Long, SampleDataLength As Long, FormatType As TxFormatType, pFormat As Long, pBitmapInfoHeader As Long, SampleStartTime As



Double, SampleStopTime As Double)

### Description

This [TOnRawSample](#) gives access to the current video sample

### See Also

[TOnRawSample](#) [OnRawAudioSample](#) [RawAudioSampleCapture](#) [RawCaptureAsyncEvent](#)  
[RawVideoSampleCapture](#)

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## OnRecordingCompleted

### TVideoGrabber.OnRecordingCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when recording ends.

### Declaration

**property** OnRecordingCompleted: TOnRecordingCompleted **read** FOnRecordingCompleted **write** FOnRecordingCompleted;

\_\_property TOnRecordingCompleted OnRecordingCompleted=read=FOnRecordingCompleted, write=FOnRecordingCompleted

Event OnRecordingCompleted(FileName As String, Success As Boolean)

### Description

This [TOnRecordingCompleted](#) event occurs after a [StopRecording](#) call, when the recording process is completed (successfully or not).

This event returns the name of the AVI file created, if any.

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with HelpNDoc's Stunning User Interface](#)

## OnRecordingPaused

### TVideoGrabber.OnRecordingPaused

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Notifies a recording paused

### Declaration

**property** OnRecordingPaused: TOnEventNotification **read** FOnRecordingPaused **write** FOnRecordingPaused;

\_\_**property** TOnEventNotification OnRecordingPaused=read=FOnRecordingPaused, write=FOnRecordingPaused

Event OnRecordingPaused

### Description

Occurs when the recording went in a paused state, e.g. after calling [PauseRecording](#).

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Markdown Content with HelpNDoc](#)

## OnRecordingReadyToStart

### TVideoGrabber.OnRecordingReadyToStart

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the recording is ready to start.

### Declaration

**property** OnRecordingReadyToStart: TEventNotification **read** FOnRecordingReadyToStart **write** FOnRecordingReadyToStart;

\_\_**property** TEventNotification OnRecordingReadyToStart=read=FOnRecordingReadyToStart, write=FOnRecordingReadyToStart

Event OnRecordingReadyToStart()

### Description

This [TOnRecordingReadyToStart](#) event allows you to start the recording at a given accurate time.

When calling [StartRecording](#) , a delay is necessary to create/open the recording file and build the recording graph.

When these operations are completed, and before the recording starts, this event occurs and holds the recording until exiting.

At this point:

- you can show a dialog asking to start or cancel recording (*see the MainDemo project from sample code*).
- invoke [ResumeRecording\(\)](#) to start the recording, or [StopRecording\(\)](#) to cancel.

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#)

[TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#)  
[AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#)  
[AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#)  
[OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)  
[OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#)  
[OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingStarted](#) [OnReencodingCompleted](#)  
[OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#)  
[PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#)  
[RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#)  
[RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#)  
[RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#)  
[SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#)  
[StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Full-featured Documentation generator

## OnRecordingStarted

### TVideoGrabber.OnRecordingStarted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the recording starts writing audio/video streams to disk.

#### Declaration

**property** OnRecordingStarted: TOnFileNotification **read** FOnRecordingStarted **write** FOnRecordingStarted;

\_\_property TOnFileNotification OnRecordingStarted=read=FOnRecordingStarted, write=FOnRecordingStarted

Event OnRecordingStarted(FileName As String)

#### Description

This [TOnFileNotification](#) event occurs when the recording starts writing audio/video streams to disk.

When calling [StartRecording](#) , a delay is necessary to create/open the recording file and build the recording graph. After all these operations are completed, this event occurs when the recording starts saving to disk.

#### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDDeinterlaceMode](#) [TAutoFileName](#)  
[TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#)  
[AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#)  
[AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last](#) [Recording](#) [FileName](#)  
[OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#)  
[OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#)  
[OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnReencodingCompleted](#)  
[OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#)  
[PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#)  
[RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#)  
[RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#)  
[RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#)  
[SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#)  
[StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Transform Your Documentation Workflow with HelpNDoc's Intuitive UI

## OnReencodingCompleted

### TVideoGrabber.OnReencodingCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when AVI reencodingion begins.

#### Declaration

**property** OnReencodingCompleted: TOnSourceFileToDestFileCompleted **read** FOnReencodingCompleted **write** FOnReencodingCompleted;

\_\_property TOnSourceFileToDestFileCompleted

OnReencodingCompleted=read=FOnReencodingCompleted, write=FOnReencodingCompleted

Event OnReencodingCompleted(SourceFile As String, DestFile As String, Success As Boolean)

#### Description

This [TOnSourceFileToDestFileCompleted](#) event occurs after stopping recording, when AVI reencodingion begins.

Note: [CompressionMode](#) must be set to cm\_CompressAfterRecording.

#### See Also

[OnReencodingProgress](#) [OnReencodingStarted](#) [Reencoding](#) [IncludeAudioStream](#) [Reencoding](#) [IncludeVideoStream](#) [Reencoding](#) [Method](#) [Reencoding](#) [NewVideoClip](#) [Reencoding](#) [SourceVideoClip](#) [Reencoding](#) [StartFrame](#) [Reencoding](#) [StartTime](#) [Reencoding](#) [StopFrame](#) [Reencoding](#) [StopTime](#) [Reencoding](#) [UseAudioCompressor](#) [Reencoding](#) [UseFrameGrabber](#) [Reencoding](#) [UseVideoCompressor](#) [Reencoding](#) [WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## OnReencodingProgress

### TVideoGrabber.OnReencodingProgress

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs periodically during AVI reencodingion.

#### Declaration

**property** OnReencodingProgress: TOnProgress **read** FOnReencodingProgress **write** FOnReencodingProgress;

\_\_property TOnProgress OnReencodingProgress=read=FOnReencodingProgress, write=FOnReencodingProgress

Event OnReencodingProgress(Percent As Long, Position As Double, Duration As Double)

#### Description

This [TOnProgress](#) event occurs periodically during AVI reencoding.

Note: [CompressionMode](#) must be set to cm\_CompressAfterRecording.

#### See Also

[OnReencodingCompleted](#) [OnReencodingStarted](#) [Reencoding](#) [IncludeAudioStream](#) [Reencoding](#) [IncludeVideoStream](#) [Reencoding](#) [Method](#) [Reencoding](#) [NewVideoClip](#) [Reencoding](#) [SourceVideoClip](#) [Reencoding](#) [StartFrame](#) [Reencoding](#) [StartTime](#) [Reencoding](#) [StopFrame](#) [Reencoding](#) [StopTime](#) [Reencoding](#) [UseAudioCompressor](#) [Reencoding](#) [UseFrameGrabber](#) [Reencoding](#) [UseVideoCompressor](#) [Reencoding](#) [WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

## OnReencodingStarted

### TVideoGrabber.OnReencodingStarted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when AVI reencodingion ends.

#### Declaration

**property** OnReencodingStarted: TOnSourceFileToDestFileStarted **read** FOnReencodingStarted **write** FOnReencodingStarted;

\_\_property TOnSourceFileToDestFileStarted OnReencodingStarted=read=FOnReencodingStarted, write=FOnReencodingStarted

Event OnReencodingStarted(SourceFile As String, DestFile As String)

#### Description

This [TOnSourceFileToDestFileStarted](#) event occurs after stopping recording, when AVI reencodingion ends. Note: [CompressionMode](#) must be set to cm\_CompressAfterRecording.

#### See Also

[OnReencodingCompleted](#) [OnReencodingProgress](#) [Reencoding](#) [IncludeAudioStream](#) [Reencoding](#) [IncludeVideoStream](#) [Reencoding](#) [Method](#) [Reencoding](#) [NewVideoClip](#) [Reencoding](#) [SourceVideoClip](#) [Reencoding](#) [StartFrame](#) [Reencoding](#) [StartTime](#) [Reencoding](#) [StopFrame](#) [Reencoding](#) [StopTime](#) [Reencoding](#) [UseAudioCompressor](#) [Reencoding](#) [UseFrameGrabber](#) [Reencoding](#) [UseVideoCompressor](#) [Reencoding](#) [WMVOutput](#) [StartReencoding](#) [StopReencoding](#)

## OnReinitializing

### TVideoGrabber.OnReinitializing

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when preview, recording or playback starts.

#### Declaration

**property** OnReinitializing: TEventNotification **read** FOnReinitializing **write** FOnReinitializing;

\_\_property TEventNotification OnReinitializing=read=FOnReinitializing, write=FOnReinitializing

Event OnReinitializing()

#### Description

Occurs when preview, recording or playback starts.

## OnResizeVideo

### TVideoGrabber.OnResizeVideo

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the video source size changes.

#### Declaration

**property** OnResizeVideo: TOnResizeVideo **read** FOnResizeVideo **write** FOnResizeVideo;

**\_\_property** TOnResizeVideo OnResizeVideo=read=FOnResizeVideo, write=FOnResizeVideo

Event OnResizeVideo(SourceWidth As Long, SourceHeight As Long)

#### Description

Unlike OnResize, that occurs when the video window is resized, this [TOnResizeVideo](#) event occurs when the video source size changes, even if [Display\\_AutoSize](#) is disabled.

#### See Also

[TOnFrameProgress](#) [TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display\\_AutoSize](#) [Display\\_FullScreen](#) [Display\\_SetLocation](#) [Display\\_VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## OnTextOverlayScrollingCompleted

### TVideoGrabber.OnTextOverlayScrollingCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a text overlay scrolling is completed

#### Declaration

**property** OnTextOverlayScrollingCompleted: TOnTextOverlayScrollingCompleted **read** FOnTextOverlayScrollingCompleted **write** FOnTextOverlayScrollingCompleted

**\_\_property** TOnTextOverlayScrollingCompleted  
OnTextOverlayScrollingCompleted=read=FOnTextOverlayScrollingCompleted,  
write=FOnTextOverlayScrollingCompleted

#### Description

This [TOnTextOverlayScrollingCompleted](#) event occurs when a text overlay scrolling is completed (when the last character has disappeared from the video frame).

The **TextOverlayIndex** parameter returns the index of the text overlay concerned (initially selected by the [TextOverlay\\_Selector](#) index)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Efficiency with a Help Authoring Tool](#)

## OnThirdPartyFilterConnected

### TVideoGrabber.OnThirdPartyFilterConnected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Unknown interface of the [third-party filter](#) when it connects

#### Declaration

**property** OnThirdPartyFilterConnected: TOnThirdPartyFilterConnected **read** FOnThirdPartyFilterConnected **write** FOnThirdPartyFilterConnected;



\_\_property TOnThirdPartyFilterConnected

OnThirdPartyFilterConnected=read=FOnThirdPartyFilterConnected, write=FOnThirdPartyFilterConnected

n/a

### Description

This [TOnThirdPartyFilterConnected](#) event returns the IUnknown interface of a [third-party filter](#) when it connects, allowing to set/retrieve the filter properties programmatically.

### See Also

[TThirdPartyFilterList](#) [TOnThirdPartyFilterConnected](#) [ThirdPartyFilter](#) [AddToList](#) [ThirdPartyFilter](#) [ClearList](#) [ThirdPartyFilter](#) [Enable](#) [ThirdPartyFilter](#) [RemoveFromList](#) [ThirdPartyFilter](#) [ShowDialog](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## OnThreadSync

### TVideoGrabber.OnThreadSync

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when the component needs to set or unset the video window parent.

### Declaration

TOnThreadSync=**procedure** (Sender: TObject; ThreadSyncPoint: TThreadSyncPoint) **of object**;

**typedef void \_\_fastcall** (\_\_closure \*TOnThreadSync)(System::TObject\* Sender, TThreadSyncPoint ThreadSyncPoint);

Event OnThreadSync

### Description

Occurs in [threaded mode](#) when the component needs to set or unset the video window parent.

### See Also

[Opening a clip or an IP URL from a background thread without blocking the main thread](#) [Player features](#) [TOnThreadSync](#) [TThreadSyncPoint](#) [EnableThreadMode](#)

Created with the Standard Edition of HelpNDoc: [Experience the power of a responsive website for your documentation](#)

## OnTVChannelScanCompleted

### TVideoGrabber.OnTVChannelScanCompleted

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Notifies the end of the TV channel scanning.

### Declaration

**property** OnTVChannelScanCompleted: TEventNotification **read** FOnTVChannelScanCompleted **write** FOnTVChannelScanCompleted;

\_\_property TEventNotification OnTVChannelScanCompleted=read=FOnTVChannelScanCompleted, write=FOnTVChannelScanCompleted

Event OnTVChannelScanCompleted()

### Description

Occurs when the TV channel scanning ends, after invoking [TVStopAutoScan](#).



This event reports the minimal and maximal channel that will be scanned for the current country code and tuner input type.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

## OnTVChannelScanStarted

### TVideoGrabber.OnTVChannelScanStarted

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Events](#)

Notifies the beginning of the TV channel scanning.

#### Declaration

**property** OnTVChannelScanStarted: TOnTVChannelScanStarted **read** FOnTVChannelScanStarted **write** FOnTVChannelScanStarted;

\_\_property TOnTVChannelScanStarted OnTVChannelScanStarted=read=FOnTVChannelScanStarted, write=FOnTVChannelScanStarted

Event OnTVChannelScanStarted(MinChannel As Long, MaxChannel As Long)

#### Description

This [TOnTVChannelScanStarted](#) event occurs when the TV channel scanning begins, after invoking [TVStartAutoScan](#).

This event reports the minimal and maximal channel that will be scanned for the current country code and tuner input type.

#### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

## OnTVChannelSelected

### TVideoGrabber.OnTVChannelSelected

[Prev](#)
[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a TV channel is selected.

#### Declaration

**property** OnTVChannelSelected: TOnTVChannelSelected **read** FOnTVChannelSelected **write** FOnTVChannelSelected;

\_\_property TOnTVChannelSelected OnTVChannelSelected=read=FOnTVChannelSelected, write=FOnTVChannelSelected

Event OnTVChannelSelected(Channel As Long, Locked As Boolean, DefaultVideoFrequency As Long, OverriddenVideoFrequency As Long, TunerVideoFrequency As Long, TunerAudioFrequency As Long)

### Description

This [TOnTVChannelSelected](#) event occurs:

- when a TV channel is selected by assigning a value to [TVChannel](#),
- for each TV channel scanned, after invoking [TVStartAutoScan](#).

See the type of this event for more information: [TOnTVChannelSelected](#).

### See Also

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: Effortlessly create a professional-quality documentation website with HelpNDoc

## OnVideoCompressionSettings

### TVideoGrabber.OnVideoCompressionSettings

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Returns capabilities about the [current video compressor](#).

### Declaration

**property** OnVideoCompressionSettings: TOnVideoCompressionSettings **read**  
FOnVideoCompressionSettings **write** FOnVideoCompressionSettings;

\_\_property TOnVideoCompressionSettings

OnVideoCompressionSettings=read=FOnVideoCompressionSettings, write=FOnVideoCompressionSettings

Event OnVideoCompressionSettings(CanKeyFrameRate As Boolean, CanPFFramesPerKeyFrame As Boolean, CanQuality As Boolean, CanWindowSize As Boolean, DefaultFrameRate As Long, DefaultPFFramesPerKey As Long, DefaultQuality As Double)

### Description

This [TOnTVChannelSelected](#) event returns the general compression capabilities of the [current video compressor](#), and its default values.

See "General video compression properties" in the [Software compression using codecs](#) chapter for more information about this event.

### See Also

[Recording methods and properties](#) [TCompressionType](#) [TOnVideoCompressionSettings](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression](#) [PFFramesPerKeyFrame](#) [VideoCompression](#) [Quality](#) [VideoCompression](#) [WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

Created with the Standard Edition of HelpNDoc: Easily create Web Help sites

## OnVideoDeviceSelected

### TVideoGrabber.OnVideoDeviceSelected

[Prev](#)

[Next](#)

[TVideoGrabber](#) [Events](#)

Occurs when a video capture device is selected.

#### Declaration

**property** OnVideoDeviceSelected: TEventNotification **read** FOnVideoDeviceSelected **write** FOnVideoDeviceSelected;

```
__property TEventNotification OnVideoDeviceSelected=read=FOnVideoDeviceSelected,
write=FOnVideoDeviceSelected
```

Event OnVideoDeviceSelected()

#### Description

Occurs when an video capture device is selected by the [VideoDevice](#) property.  
The device-dependent settings of this video capture device are reloaded and then this event occurs.

#### See Also

[WDM drivers](#) [AutoConnectRelatedPins](#) [CurrentFrameRate](#) [DeliveredFrames](#) [DroppedFrames](#) [FixFlickerOrBlackCapture](#) [IsDVDevice](#) [IsVideoControlAvailable](#) [IsVideoCrossbarAvailable](#) [IsVideoDeviceConnected](#) [IsWDMVideoDriver](#) [OnNoVideoDevices](#) [RefreshDevicesAndCompressorsLists](#) [ResetVideoDeviceSettings](#) [ShowDialog](#) [VideoDevice](#) [VideoDeviceIndex](#) [VideoDeviceIndexFromId](#) [VideoDeviceName](#) [VideoDevices](#) [VideoDevicesCount](#) [VideoDevicesId](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## OnVideoFromBitmapsNextFrameNeeded

### TVideoGrabber.OnVideoFromBitmapsNextFrameNeeded

[Prev](#)

[TVideoGrabber](#) [Events](#)

Requests a bitmap handle to build the next video frame.

#### Declaration

**property** OnVideoFromBitmapsNextFrameNeeded: TOnVideoFromBitmapsNextFrameNeeded **read** FOnVideoFromBitmapsNextFrameNeeded **write** FOnVideoFromBitmapsNextFrameNeeded;

```
__property TOnVideoFromBitmapsNextFrameNeeded
OnVideoFromBitmapsNextFrameNeeded=read=FOnVideoFromBitmapsNextFrameNeeded,
write=FOnVideoFromBitmapsNextFrameNeeded
```

Event OnVideoFromBitmapsNextFrameNeeded(FirstSample As Boolean, BitmapHandle, BMPorJPEGFile, EndOfData)

#### Description

This [TOnVideoFromBitmapsNextFrameNeeded](#) event occurs for each video frame that will be build from a bitmap handle when a preview or recording graph is running and [VideoSource](#) = vs\_JPEGsOrBitmaps.

**From this event you must invoke** [SendImageToVideoFromBitmaps](#), and pass as parameter:

- either a file path to a BMP or JPEG file through the **BMPorJPEGFile** parameter (a)
- either a bitmap handle through the **BitmapHandle** parameter (b)

If no more images are available, set the **EndOfData** parameter to true to notify the end of the recording or preview.

*Remarks:*

- (a) if a BMPorJPEGFile string is specified (the string is not empty), set the BitmapHandle parameter to 0
- (b) if a bitmap handle is used, enable **CanFreeBitmapHandle** if TVideoGrabber must free the bitmap handle, or disable **CanFreeBitmapHandle** if you need to reuse the bitmap later.

*You can find sample code in the MainDemo project included in the package.*

**See Also**

[TOnVideoFromBitmapsNextFrameNeeded](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Review with HelpNDoc's Project Analyzer](#)

## Types

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

### TAero

#### TAero

[Prev](#)

[Next](#)

Aero type

**Declaration**

TAero=(ae\_Default, ae\_AutoBestRenderingQuality, ae\_ForceOnWhenStartingVideo, ae\_ForceOffWhenStartingVideo, ae\_ForceOnImmediately, ae\_ForceOffImmediately);

**enum** TAero ae\_Default, ae\_AutoBestRenderingQuality, ae\_ForceOnWhenStartingVideo, ae\_ForceOffWhenStartingVideo, ae\_ForceOnImmediately, ae\_ForceOffImmediately ;

**Description**

Type of the Aero property

ae\_Default: does nothing

ae\_AutoBestRenderingQuality: automatically disable Aero if required for the best video quality

ae\_ForceOnWhenStartingVideo: re-enable aero (if disabled) when starting the video

ae\_ForceOffWhenStartingVideo: disable aero (if enabled) when starting the video

ae\_ForceOnImmediately: re-enable aero immediately

ae\_ForceOffImmediately: disable aero immediately

Created with the Standard Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

### TApplicationPriority

#### TApplicationPriority

[Prev](#)

[Next](#)

Type of the [ApplicationPriority](#) property.

**Description**

Type of the [ApplicationPriority](#) property.

**Values**

current priority

idle priority

normal priority

high priority

ap\_default: (usually ap\_normal value)  
ap\_normal: ap\_high:  
ap\_idle:

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

## TASFDeinterlaceMode

### TASFDeinterlaceMode

[Prev](#)

[Next](#)

Type of the [ASFDeinterlaceMode](#) property.

**Unit**  
[VidGrab](#)

#### Declaration

TASFDeinterlaceMode=(adm\_NotInterlaced, adm\_DeinterlaceNormal, adm\_DeinterlaceHalfSize, adm\_DeinterlaceHalfSizeDoubleRate, adm\_DeinterlaceInverseTelecine, adm\_DeinterlaceVerticalHalfSizeDoubleRate);

**enum** TASFDeinterlaceMode adm\_NotInterlaced, adm\_DeinterlaceNormal, adm\_DeinterlaceHalfSize, adm\_DeinterlaceHalfSizeDoubleRate, adm\_DeinterlaceInverseTelecine, adm\_DeinterlaceVerticalHalfSizeDoubleRate ;

#### Description

Type of the [ASFDeinterlaceMode](#) property.

#### Possible values:

adm\_NotInterlaced  
adm\_DeinterlaceNormal  
adm\_DeinterlaceHalfSize  
adm\_DeinterlaceHalfSizeDoubleRate  
adm\_DeinterlaceInverseTelecine  
adm\_DeinterlaceVerticalHalfSizeDoubleRate

#### See Also

Created with the Standard Edition of HelpNDoc: [Revolutionize your documentation process with HelpNDoc's online capabilities](#)

## TASFProfileVersion

### TASFProfileVersion

[Prev](#)

[Next](#)

Type of the [ASFProfileVersion](#) property.

#### Description

Allowed values:

apv\_ProfileVersion\_8: use the WMV profiles version 8  
apv\_ProfileVersion\_9: use the WMV profiles version 9

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## TAspectRatio

### TAspectRatio

[Prev](#)

[Next](#)

Type of the [Display\\_AspectRatio](#) and [DualDisplay\\_AspectRatio](#) properties.  
**Description**

**ar\_Box:** the video appears in full within the control in a letterbox (bars on the top and the bottom) or pillarbox (bars on the left and the right)

**ar\_NoResize:** the video appears in its "native" size. If the native size is larger than then control, it appears centered within the control. If the native size is larger than the control, the video appears truncated.

**ar\_Stretch:** the video is stretched as needed to fill the control

**ar\_PanScan:** the video may be panned or scanned, according to the [Display\\_PanScanRatio](#) value (in the 0..100 range, default value 50).

---

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

---

## TAudioSource

### TAudioSource

[Prev](#)

[Next](#)

Type of the [AudioSource](#) property.

#### Declaration

TAudioSource=(as\_Default, as\_UseExternalAudio);

**enum** TAudioSource as\_Default, as\_UseExternalAudio ;

#### Description

Type of the [AudioSource](#) property.

Possible values:

**as\_Default:**

uses the audio output of the [video capture device](#) if any, otherwise uses the current [audio capture device](#).

**as\_UseExternalAudio:**

**always uses the current [audio capture device](#) for the audio recording, even if the current source (e.g. an URL) exposes an audio output.**

**as\_SpeakerOutput:**

uses the soundboard's audio output (captures whatever is rendered)

**as\_DefaultWithSpeakerRecording:** (during recording only)

if AudioDeviceRendering is enabled, the audio of the current source is rendered, but the soundboard's audio output is recorded

**as\_silent**

generates an "empty" audio stream

**as\_IPCamera**

allows to use an IP camera as audio source when previewing or recording an USB source, the screen or the video mixer (when VideoSource = vs\_VideoCaptureDevice, vs\_ScreenRecording or vs\_Mixer).

VideoGrabber.IPCameraURL and eventually the VideoGrabber.SetAuthentication (at\_IPCamera, ...) must be set accordingly.

---

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

---

## TAuthenticationType

### TAuthenticationType

[Prev](#)

[Next](#)

Authentication type.

## Declaration

TAuthenticationType=(at\_PublishingPoint, at\_IPCamera, at\_StreamingUrl);

**enum** TAuthenticationType at\_PublishingPoint, at\_IPCamera, at\_StreamingUrl ;

## Description

Type used by the [SetAuthentication](#) method and the [OnAuthenticationNeeded](#) event.

Created with the Standard Edition of HelpNDoc: [Full-featured Documentation generator](#)

## TAutoFileName

### TAutoFileName

[Prev](#)

[Next](#)

Type used by the [AutoFileName](#) property.

## Unit

[VidGrab](#)

## Declaration

TAutoFileName=(fn\_Sequential, fn\_DateTime);

**enum** TAutoFileName fn\_Sequential, fn\_DateTime ;

## Description

### fn\_Sequential:

The file name is based on a sequential number prefixed by [AutoFilePrefix](#) , starting from 000001.

e.g.: vg000010.avi, vg000011.avi

fn\_DateTime:

The file name is based on the yymmdd\_hhmmss\_zzz date/time format (zzz = milliseconds) prefixed by [AutoFilePrefix](#) .

e.g. vg030616\_090603\_904.jpg

### fn\_Counter:

equivalent to **fn\_Sequential**, but existing files are overwritten.

### fn\_Modulo:

used to capture the last N frames.

If you set BurstMode = true and BurstCount = N, the last N frames are generated, and when N is reached the file number restarts from 1.

E.g. if BurstCount = 5, the files are written in the following order:

vg000001.jpg vg000002.jpg vg000003.jpg vg000004.jpg vg000005.jpg **vg000001.jpg vg000002.jpg vg000003.jpg ...**

## See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TFrameCaptureDest](#) [TOnFrameCaptureCompleted](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [Encoder](#) [SetInt](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [HoldRecording](#) [IsRecordingPaused](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last\\_BurstFrameCapture](#) [FileName](#) [Last\\_CaptureFrameTo](#) [FileName](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnFrameBitmap](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#)



[RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetFrameCaptureBounds](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Documentation with a Help Authoring Tool](#)

## TAVIInfoType

### TAVIInfoType

[Prev](#)

[Next](#)

Type used by the AVIInfo2 function

#### Description

Type used to select the string returned by the AVIInfo2 function

Possible values:

av\_Duration  
av\_FrameCount  
av\_VideoWidth  
av\_VideoHeight  
av\_VideoFrameRateFps  
av\_VideoCodec  
av\_AudioCodec  
av\_AvgBitRate  
av\_AudioChannels  
av\_AudioSamplesPerSec  
av\_AudioBitsPerSample  
av\_FileSizeInKB  
av\_AudioStreams

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## TAVIMuxConfig

### TAVIMuxConfig

[Prev](#)

[Next](#)

TAVIMuxConfig type

#### Declaration

TAVIMuxConfig=(avmx\_SetInterleavingMode, avmx\_SetInterleave, avmx\_SetPreroll);

**enum** TAVIMuxConfig avmx\_SetInterleavingMode, avmx\_SetInterleave, avmx\_SetPreroll ;

#### Description

Type used by [SetAVIMuxConfig](#)

#### See Also

[Recording methods and properties](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#)

[OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#)  
[OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#)  
[PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#)  
[RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#)  
[RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#)  
[RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#)  
[SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#)  
[StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Easily create PDF Help documents](#)

## TCardinalDirection

### TCardinalDirection

[Prev](#)

[Next](#)

Type used by the [TextOverlay\\_ShadowDirection](#) property.

#### Declaration

```
TCardinalDirection=(cd_North, cd_NorthEast, cd_East, cd_SouthEast, cd_South, cd_SouthWest,
cd_West, cd_NorthWest, cd_Center);
```

```
enum TCardinalDirection _North, cd_NorthEast, cd_East, cd_SouthEast, cd_South, cd_SouthWest,
cd_West, cd_NorthWest, cd_Center;
```

#### Description

Type used by the [TextOverlay\\_ShadowDirection](#) property.

The value can be one of the following:

```
cd_North
cd_NorthEast
cd_East
cd_SouthEast
cd_South
cd_SouthWest
cd_West
cd_NorthWest
cd_Center
```

#### See Also

[TOnFrameOverlayUsingDC](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#)  
[DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEONDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)

[TextOverlay\\_CreateCustomFont](#) [TextOverlay\\_CreateCustomFont2](#) [TextOverlay\\_Enabled](#) [TextOverlay\\_Font](#) [TextOverlay\\_FontColor](#) [TextOverlay\\_Left](#) [TextOverlay\\_Right](#) [TextOverlay\\_Scrolling](#) [TextOverlay\\_ScrollingSpeed](#) [TextOverlay\\_Selector](#) [TextOverlay\\_Shadow](#) [TextOverlay\\_ShadowColor](#) [TextOverlay\\_ShadowDirection](#) [TextOverlay\\_String](#) [TextOverlay\\_Top](#) [TextOverlay\\_Transparent](#) [TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## TDialog

### TDialog

[Prev](#)

[Next](#)

Type used by the [ShowDialog](#) function.

#### Unit

[VidGrab](#)

#### Declaration

```
TDialog=(dlg_VideoDevice, dlg_VideoCompressor, dlg_AudioCompressor, dlg_StreamConfig,
dlg_VfwFormat, dlg_VfwSource, dlg_vfwDisplay, dlg_VideoCrossbar, dlg_AudioCrossbar, dlg_TVTuner,
dlg_TVAudio, dlg_AudioDevice, dlg_NetShowConfig, dlg_DScaler, dlg_FFDSHOWVideo,
dlg_FFDSHOWAudio);
```

```
enum TDialog _VideoDevice, dlg_VideoCompressor, dlg_AudioCompressor, dlg_StreamConfig,
dlg_VfwFormat, dlg_VfwSource, dlg_vfwDisplay, dlg_VideoCrossbar, dlg_AudioCrossbar, dlg_TVTuner,
dlg_TVAudio, dlg_AudioDevice, dlg_NetShowConfig, dlg_DScaler, dlg_FFDSHOWVideo,
dlg_FFDSHOWAudio;
```

#### Description

Type used by the [ShowDialog](#) function.

#### Value Meaning

**dlg\_VideoDevice:** dialog of the [video capture device](#) ,  
**dlg\_VideoCompressor:** dialog of the [video compressor](#) , if any,  
**dlg\_AudioCompressor:** dialog of the [audio compressor](#) , if any,  
**dlg\_StreamConfig:** dialog of the video capture stream,  
**dlg\_VfwFormat:** format dialog of the video capture device ([vfw](#) drivers only),  
**dlg\_VfwSource:** source dialog of the video capture device ([vfw](#) drivers only),  
**dlg\_vfwDisplay:** source dialog of the video capture device ([vfw](#) drivers only),  
**dlg\_VideoCrossbar:** dialog of the video crossbar, if any,  
**dlg\_AudioCrossbar:** dialog of the audio crossbar, if any,  
**dlg\_TVTuner:** dialog of the [TV tuner](#) , if any,  
**dlg\_TVAudio:** dialog of the TV audio interface, if any,  
**dlg\_AudioDevice:** dialog of the [audio capture device](#)  
**dlg\_NetShowConfig:** dialog of the streaming source (when playing from an URL)  
**dlg\_DScaler:** dialog of the DScaler deinterlacer (see [DScaler deinterlacer](#))  
**dlg\_FFDSHOWVideo:** dialog of the [FFDSHOW Video Decoder](#)  
**dlg\_FFDSHOWAudio:** dialog of the [FFDSHOW Audio Decoder](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## TDiscoveryCallbackStatus

### TDiscoveryCallbackStatus

[Prev](#)

[Next](#)

TDiscoveryCallbackStatus type

#### Description

TDiscoveryCallbackStatus = (dcs\_CameraFound, dcs\_MulticastCompleted, dcs\_IPRangeCompleted);  
 This type is returned the the [OnONVIFDiscoveryCompletedNotification](#) event.

## See Also

[IR Cut Filter of Axis cameras](#) [TOnONVIFDiscoveryCompletedNotification](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastError](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF](#) [GetStr](#) [ONVIF](#) [SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [News and information about help authoring tools and software](#)

## TDVCommand

### TDVCommand

[Prev](#)

[Next](#)

Type used by the DV camcoders functions.

### Unit

[VidGrab](#)

### Declaration

TDVCommand=(dv\_Play, dv\_Stop, dv\_Freeze, dv\_Thaw, dv\_Ff, dv\_Rew, dv\_Record, dv\_RecordFreeze, dv\_RecordStrobe, dv\_StepFwd, dv\_StepRev, dv\_ModeShuttle, dv\_PlayFastestFwd, dv\_PlaySlowestFwd, dv\_PlayFastestRev, dv\_PlaySlowestRev);

**enum** TDVCommand dv\_Play, dv\_Stop, dv\_Freeze, dv\_Thaw, dv\_Ff, dv\_Rew, dv\_Record, dv\_RecordFreeze, dv\_RecordStrobe, dv\_StepFwd, dv\_StepRev, dv\_ModeShuttle, dv\_PlayFastestFwd, dv\_PlaySlowestFwd, dv\_PlayFastestRev, dv\_PlaySlowestRev ;

### Description

Type used by the DV camcoders functions.

#### Values:

dv\_Play  
dv\_Stop  
dv\_Freeze  
dv\_Thaw  
dv\_Ff  
dv\_Rew  
dv\_Record  
dv\_RecordFreeze  
dv\_RecordStrobe  
dv\_StepFwd  
dv\_StepRev  
dv\_ModeShuttle  
dv\_PlayFastestFwd  
dv\_PlaySlowestFwd  
dv\_PlayFastestRev  
dv\_PlaySlowestRev

## See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#)

[DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## TDVDInfoType

### TDVDInfoType

[Prev](#)

[Next](#)

TDVDInfo type

#### Declaration

TDVDInfoType=(dvd\_NumberOfVolumes, dvd\_TotalDuration, dvd\_NumberOfTitles, dvd\_TitleDuration, dvd\_TitleFrameRate, dvd\_SourceResolutionX, dvd\_SourceResolutionY, dvd\_TitleFrameCount)

**enum** TDVDInfoType dvd\_NumberOfVolumes, dvd\_TotalDuration, dvd\_NumberOfTitles, dvd\_TitleDuration, dvd\_TitleFrameRate, dvd\_SourceResolutionX, dvd\_SourceResolutionY, dvd\_TitleFrameCount ;

#### Description

Type used by the [DVDInfo](#) function.

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

## TDVSize

### TDVSize

[Prev](#)

[Next](#)

Type of the [PlayerDVSize](#) property.

#### Unit

[VidGrab](#)

#### Declaration

TDVSize=(dv\_Full, dv\_Half, dv\_Quarter, dv\_DC);

**enum** TDVSize dv\_Full, dv\_Half, dv\_Quarter, dv\_DC ;

#### Description

Type of the [PlayerDVSize](#) property.

#### Value Meaning

dvFull full size  
dvHalf half size  
dvQuarter quarter size  
dvDC 1/8 size

#### See Also

[TDVVideoFormat](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with a Help Authoring Tool](#)

## TDVVideoFormat

### TDVVideoFormat

[Prev](#)

[Next](#)

Type of DV Format.

#### Unit

[VidGrab](#)

#### Declaration

TDVVideoFormat=(dvf\_Default, dvf\_DVSD, dvf\_DVHD, dvf\_DVSL);

**enum** TDVVideoFormat dvf\_Default, dvf\_DVSD, dvf\_DVHD, dvf\_DVSL ;

#### Description

##### Value

dvDVSD

dvDVHD

dvDVSL

#### See Also

[TDVSize](#) [TDVVideoStandard](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#)  
[DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#)  
[DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#)  
[OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

## TDVVideoStandard

### TDVVideoStandard

[Prev](#)

[Next](#)

Type of DV standard.

#### Unit

[VidGrab](#)

#### Declaration

TDVVideoStandard=(dvs\_Default, dvs\_PAL, dvs\_NTSC);

**enum** TDVVideoStandard dvs\_Default, dvs\_PAL, dvs\_NTSC ;

#### Description

##### Value

dvPAL

dvNTSC

#### See Also

[TDVSize](#) [TDVVideoFormat](#) [TOnDVCommandCompleted](#) [DVDateTimeEnabled](#)  
[DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#)  
[DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#)  
[OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## TEncoder\_int

### TEncoder\_int

[Prev](#)[Next](#)

TEncoder\_int

#### Declaration

```

TEncoder_int=(
    Enc_IsActive_bool,
    Enc_Bytes_Written_kb_readonly,
    Enc_Audio_Enabled_bool,
    Enc_Audio_Channels,
    Enc_Audio_SamplesPerSec,
    Enc_Audio_BitsPerSample,
    Enc_Audio_BitRate_kb,
    Enc_Audio_PCM_Format,
    Enc_Video_Enabled_bool,
    Enc_Video_Width,
    Enc_Video_Height,
    Enc_Video_BitCount,
    Enc_Video_AvgTimePerFrame,
    Enc_Video_BitRate_kb,
    Enc_Video_rc_MinBitRate_kb,
    Enc_Video_rc_MaxBitRate_kb,
    Enc_Video_rc_BufferSize_kb,
    Enc_Video_IDR_Interval,
    Enc_Video_Max_BFrames,
    Enc_Video_FrameRate_x100,
    Enc_IsRealTime_bool,
    Enc_IsScreenRecording_bool,
    Enc_Video_Thread_Count,
    Enc_Video_GPU_Encoder,
    Enc_Video_GPU_EncoderDevice

```

```

enum TEncoder_int Enc_IsActive, Enc_Bytes_Written_kb_readonly, Enc_Audio_Enabled,
Enc_Audio_Channels, Enc_Audio_SamplesPerSec, Enc_Audio_BitsPerSample, Enc_Audio_BitRate_kb,
Enc_Video_Enabled, Enc_Video_Width, Enc_Video_Height, Enc_Video_BitRate_kb,
Enc_Video_rc_MinBitRate_kb, Enc_Video_rc_MaxBitRate_kb, Enc_Video_rc_BufferSize_kb,
Enc_Video_IDR_Interval, Enc_Video_Max_BFrames, Enc_Video_FrameRate_x100,
Enc_Video_Thread_Count, Enc_Video_GPU_Encoder, Enc_Video_GPU_Encoder_DeviceNumber ;

```

**Enc\_Video\_GPU\_Encoder** possible values (pass the index as integer, in the 0..n-1 range):

```

TGPUEncoder=(
    Enc_GPU_None,
    Enc_GPU_Auto,
    Enc_GPU_Intel_QSV,

```



Enc\_GPU\_NVidia\_NVENC,  
Enc\_GPU\_AMD\_AMF

**Enc\_Audio\_PCM\_Format** possible values (pass the index as integer, in the 0..n-1 range):

```
TAudioPCMFormat=(
PCM_U8,
PCM_S16,
PCM_S32,
PCM_FLT,
PCM_DBL
```

#### Description

The TEncoder\_int type is used by the [Datastead Encoder](#)

Created with the Standard Edition of HelpNDoc: [Keep Your Sensitive PDFs Safe with These Easy Security Measures](#)

## TEncoder\_str

### TEncoder\_str

[Prev](#)
[Next](#)

TEncoder\_str

#### Declaration

```
TEncoder_str=(
Enc_OutputURL,
Enc_Video_Codec,
Enc_Audio_Codec,
Enc_Extra_Parameters
```

**enum** TEncoder\_str Enc\_OutputURL, Enc\_Video\_Codec, Enc\_Audio\_Codec, Enc\_Extra\_Parameters ;

#### Description

The TEncoder\_str type is used by the [Datastead Encoder](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## TEncryptionMethod

### TEncryptionMethod

[Prev](#)
[Next](#)

Type of encryption method

#### Declaration

```
TEncryptionMethod =(em_Fast_Encryption, em_AES_Encryption);
```

**enum** TEncryptionMethod: **unsigned int** em\_Fast\_Encryption, em\_AES\_Encryption ;

#### Description

Type of encryption method:

**em\_Fast\_Encryption:** fast encryption, suitable for real-time encryption of large video sizes and/or several concurrent recordings

**em\_AES\_Encryption:** AES encryption. Safer, but slower.

#### See Also

[Decrypt\\_File](#) [Encrypt\\_File](#) [EncryptionMethod](#) [SetDecryptionKey](#) [SetEncryptionKey](#)

Created with the Standard Edition of HelpNDoc: [Add an Extra Layer of Security to Your PDFs with Encryption](#)

## TEventNotification

### TEventNotification

[Prev](#)

[Next](#)

TNotifyEvent

#### Unit

[VidGrab](#)

#### Declaration

TEventNotification=**procedure**(Sender: TObject) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TEventNotification)(System::TObject \*Sender);

#### Description

Equivalent to a TNotifyEvent type.

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

## TFileSort

### TFileSort

[Prev](#)

[Next](#)

Type of the [VideoFromImages](#) [BitmapsSortedBy](#) property.

#### Unit

[VidGrab](#)

#### Declaration

TFileSort=(fs\_TimeAsc, fs\_TimeDesc, fs\_NameAsc, fs\_NameDesc);

**enum** TFileSort fs\_TimeAsc, fs\_TimeDesc, fs\_NameAsc, fs\_NameDesc ;

#### Description

##### Value Meaning

fs\_TimeAsc the image files are sorted by time, ascending

fs\_TimeDesc the image files are sorted by time, descending

fs\_NameAsc the image files are sorted by name, ascending

fs\_NameDesc the image files are sorted by name, descending

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## TFormatType

### TFormatType

[Prev](#)

[Next](#)

Type of the format returned by the raw sample events.

#### Description

Type of the format returned by the [OnRawVideoSample](#) and [OnRawAudioSample](#) events.

Possible values:

ft\_VideoInfo,  
ft\_VideoInfo2  
ft\_DvInfo  
ft\_Mpeg1Video  
ft\_Mpeg2Video  
ft\_MpegStreamType  
ft\_MpegCustom  
ft\_WaveFormatEx  
ft\_Unknown

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## TFrameBitmapInfo

### TFrameBitmapInfo

[Prev](#)

[Next](#)

Record type returning the bitmap info of the current video frame

#### Description

Record type used by the [OnFrameBitmap](#) event.

This record contains information about the current frame bitmap, and information about current/recent mouse positions and states

```

    BitmapWidth: LongInt
    BitmapHeight: LongInt
    BitmapBitsPerPixel: LongInt
    BitmapLineSize: LongInt
    BitmapSize: LongInt
    BitmapPlanes: LongInt
    BitmapHandle: THandle           // handle of the frame bitmap DIB
    BitmapDataPtr: pByte           // pointer to the bitmap data array
    BitmapDC: HDC                  // device context of the frame bitmap
    CurrentXMouseLocation: LongInt  // current X position of the mouse
    CurrentYMouseLocation: LongInt  // current y position of the mouse
    LastXMouseDownLocation: LongInt // last X position of the mouse when the button
    LastYMouseDownLocation: LongInt // last Y position of the mouse when the button
    IsMouseDown: Boolean           // true if the mouse button is currently down
    LastMouseButtonClicked: TMouseButton // last mouse button clicked
    Reserved0: LongInt
    
```

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## TFrameCaptureDest

### TFrameCaptureDest

[Prev](#)

[Next](#)

Type used by the [BurstType](#) property and [CaptureFrameTo](#) function.

#### Unit

[VidGrab](#)

#### Declaration

TFrameCaptureDest=(fc\_TBitmap, fc\_BmpFile, fc\_JpegFile);

**enum** TFrameCaptureDest fc\_TBitmap, fc\_BmpFile, fc\_JpegFile ;

### Description

Type used by the [BurstType](#) property and [CaptureFrameTo](#) function.

### Value Meaning

**fc\_TBitmap:** the frame will be captured to to a memory bitmap (TBitmap in Delphi / C++Builder, HBitmap handle in the OCX versions) and returned by the [OnFrameCaptureCompleted](#) event.

**fc\_BmpFile:** the frame will be saved to a Bmp file, (\*)

**fc\_JpegFile:** the frame will be saved to a Jpeg file, (\*)

**fc\_Clipboard:** the frame will be saved in the clipboard in CF\_BITMAP format, it can be pasted from another application.

(\*) in both cases the [OnFrameCaptureCompleted](#) event occurs and returns also the file name of the file created. The file name generated depends of the [AutoFilePrefix](#) and [AutoFileName](#) properties.

### See Also

[TAutoFileName](#) [TOnFrameCaptureCompleted](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: Streamline your documentation process with HelpNDoc's HTML5 template

## TFrameInfo

### TFrameInfo

[Prev](#)

[Next](#)

Record type returning the current frame info

### Description

Record type used by the [OnFrameBitmap](#) and [OnFrameProgress2](#) events.

This record contains the following information about the current frame:

FrameTime: int64;  
 FrameTime\_TotalMin: int64;  
 FrameTime\_TotalSec: int64;  
 FrameTime\_TotalHs: int64;  
 FrameNumber: int64;  
 DroppedFrameCount: LongInt;  
 FrameTime\_Hour: LongInt;  
 FrameTime\_Min: LongInt;  
 FrameTime\_Sec: LongInt;  
 FrameTime\_Hs: LongInt;  
 DVTimeCode\_IsAvailable: LongInt;  
 DVTimeCode\_Hour: LongInt;  
 DVTimeCode\_Min: LongInt;  
 DVTimeCode\_Sec: LongInt;  
 DVTimeCode\_Ff: LongInt;  
 DVTimeCode\_TrackNumber: LongInt;  
 DVDDateTime\_IsAvailable: LongInt;  
 DVDDateTime\_Year: LongInt;

```

DVDDateTime_Month: LongInt;
DVDDateTime_Day: LongInt;
DVDDateTime_Hour: LongInt;
DVDDateTime_Min: LongInt;
DVDDateTime_Sec: LongInt;
CurrentState: TCurrentState;
GraphState: TGraphState;
PlayerState: TPlayerState;
PlaylistIndex: LongInt;
NTPFrameTime: int64;

```

Created with the Standard Edition of HelpNDoc: [Say Goodbye to Documentation Headaches with a Help Authoring Tool](#)

## TFrameInfoId

### TFrameInfoId

[Prev](#)
[Next](#)

Type of the frame info requested. Used with [GetFrameInfo](#).

#### Unit

[VidGrab](#)

#### Declaration

```

TFrameInfoId=(fi_FrameNumber, fi_DroppedFrameCount, fi_SampleTime_Hour, fi_SampleTime_Min,
fi_SampleTime_Sec, fi_SampleTime_Hs, fi_SampleTime_TotalMin, fi_DVTimeCode_IsAvailable,
fi_DVTimeCode_Hour, fi_DVTimeCode_Min, fi_DVTimeCode_Sec, fi_DVTimeCode_Ff,
fi_DVTimeCode_TrackNumber, fi_DVDDateTime_IsAvailable, fi_DVDDateTime_Year, fi_DVDDateTime_Month,
fi_DVDDateTime_Day, fi_DVDDateTime_Hour, fi_DVDDateTime_Min, fi_DVDDateTime_Sec);

```

```

enum TFrameInfoId fi_FrameNumber, fi_DroppedFrameCount, fi_SampleTime_Hour, fi_SampleTime_Min,
fi_SampleTime_Sec, fi_SampleTime_Hs, fi_SampleTime_TotalMin, fi_DVTimeCode_IsAvailable,
fi_DVTimeCode_Hour, fi_DVTimeCode_Min, fi_DVTimeCode_Sec, fi_DVTimeCode_Ff,
fi_DVTimeCode_TrackNumber, fi_DVDDateTime_IsAvailable, fi_DVDDateTime_Year, fi_DVDDateTime_Month,
fi_DVDDateTime_Day, fi_DVDDateTime_Hour, fi_DVDDateTime_Min, fi_DVDDateTime_Sec ;

```

#### Description

##### Value

```

fi_FrameNumber
fi_DroppedFrameCount
fi_SampleTime_Hour
fi_SampleTime_Min
fi_SampleTime_Sec
fi_SampleTime_Hs
fi_SampleTime_TotalMin
fi_DVTimeCode_IsAvailable
fi_DVTimeCode_Hour
fi_DVTimeCode_Min
fi_DVTimeCode_Sec
fi_DVTimeCode_Ff
fi_DVTimeCode_TrackNumber
fi_DVDDateTime_IsAvailable
fi_DVDDateTime_Year
fi_DVDDateTime_Month
fi_DVDDateTime_Day
fi_DVDDateTime_Hour
fi_DVDDateTime_Min
fi_DVDDateTime_Sec
fi_NTPFrameTime

```

---

Created with the Standard Edition of HelpNDoc: [Effortlessly create a professional-quality documentation website with HelpNDoc](#)

---

## TFrameInfoStringId

### TFrameInfoStringId

[Prev](#)[Next](#)

Type used by the [GetFrameInfoString](#) function

#### Description

Type used by the [GetFrameInfoString](#) function

Possible values:

```
TFrameInfoStringId = (  
    fis_DVTimeCode,  
    fis_DVDateTime,  
    fis_TimeCode,  
    fis_FrameTime,  
    fis_FrameNumber,  
    fis_FullInfo  
);
```

---

Created with the Standard Edition of HelpNDoc: [Easily convert your WinHelp HLP help files to CHM with HelpNDoc's step-by-step guide](#)

---

## TGPUEncoder

### TGPUEncoder

[Prev](#)[Next](#)

TGPUEncoder

#### Declaration

```
TGPUEncoder=(  
    Enc_GPU_None,  
    Enc_GPU_Auto,  
    Enc_GPU_Intel_QSV,  
    Enc_GPU_NVidia_NVENC,  
    Enc_GPU_AMD_AMF
```

```
enum TGPUEncoder Enc_GPU_None, Enc_GPU_Auto, Enc_GPU_Intel_QSV, Enc_GPU_NVidia_NVENC,  
    Enc_GPU_AMD_AMF ;
```

#### Description

The TGPUEncoder type is used by the [Datastead Encoder](#)

---

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

---

## TGraphState

### TGraphState

[Prev](#)[Next](#)

Type returned by the [GraphState](#) function.

**Declaration**

TGraphState=(gs\_Stopped, gs\_Paused, gs\_Running);

**enum** TGraphState gs\_Stopped, gs\_Paused, gs\_Running ;

**Description**

Type returned by the [GraphState](#) function.

Possible values:

**gs\_Stopped:** the graph is in a stopped state

**gs\_Paused:** the graph is in a paused state

**gs\_Running:** the graph is in a running state

---

Created with the Standard Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

---

## THeaderAttribute

### THeaderAttribute

[Prev](#)[Next](#)

Type of the AVI or ASF attribute that can be set by invoking [SetHeaderAttribute](#).

**Description**

See [SetHeaderAttribute](#) for the list of possible values.

**See Also**

[ClearHeaderAttributes](#) [SetHeaderAttribute](#)

---

Created with the Standard Edition of HelpNDoc: [What is a Help Authoring tool?](#)

---

## THwAccel

### THwAccel

[Prev](#)[Next](#)

Type of the [PlayerHwAccel](#) property

**Description**

Possible values:

**hw\_None:** no hardware acceleration

**hw\_Cuda:** NVidia CUDA

**hw\_QuickSync:/b0 Intel QuickSync**

**hw\_Dxva2:** DirectX dxva2

**hw\_d3d11:** DirectX d3d11

---

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

---

## TIPCameraSetting

### TIPCameraSetting

[Prev](#)[Next](#)

TIPCameraSetting

**Declaration**

TIPCameraSetting=(ips\_ConnectionTimeOut, ips\_ReceiveTimeOut);



**enum** TIPCameraSetting ips\_ConnectionTimeOut, ips\_ReceiveTimeOut);

### Description

Type used by the [SetIPCameraSetting](#) method.

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## TLogoLayout

### TLogoLayout

[Prev](#)
[Next](#)

Type of the [LogoLayout](#) property.

### Unit

[VidGrab](#)

### Declaration

TLogoLayout=(lg\_Centered, lg\_Stretched, lg\_Repeated, lg\_TopLeft, lg\_TopRight, lg\_BottomLeft, lg\_BottomRight);

**enum** TLogoLayout lg\_Centered, lg\_Stretched, lg\_Repeated, lg\_TopLeft, lg\_TopRight, lg\_BottomLeft, lg\_BottomRight ;

### Description

#### Value Meaning

lg\_Centered the logo is centered "as is" in the middle of the video window

lg\_Stretched the logo is stretched to fit the video window

lg\_Repeated the logo is repeated over the video window, until filled out

lg\_TopLeft the logo is aligned on the top left of the control

lg\_TopRight the logo is aligned on the top right of the control

lg\_BottomLeft the logo is aligned on the bottom left of the control

lg\_BottomRight the logo is aligned on the bottom right of the control

lg\_Boxed the logo is boxed to preserve its aspect ratio. Depending of its aspect ratio, borders are added on the top and bottom, or on the left and right (the border color is [BackgroundColor](#))

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## TLogType

### TLogType

[Prev](#)
[Next](#)

Constants returned by the [OnLog](#) event.

### Unit

[VidGrab](#)

### Declaration

TLogType=(e\_add\_filter, e\_add\_source\_filter, e\_audio\_compressor\_not\_suitable, e\_bind\_moniker\_to\_filter, e\_compressor\_possibly\_not\_suitable, e\_create\_instance, e\_ddraw\_caps\_not\_suitable, e\_device\_in\_use\_in\_another\_graph, e\_disk\_full, e\_failed, e\_failed\_to\_allocate\_recording\_file, e\_failed\_to\_bind\_codec, e\_failed\_to\_connect\_crossbar\_pin, e\_failed\_to\_connect\_to\_server, e\_failed\_to\_create\_directory, e\_failed\_to\_create\_file, e\_failed\_to\_create\_temp, e\_failed\_to\_bind\_frame\_grabber, e\_failed\_to\_load\_ASF\_profile, e\_failed\_to\_load\_ASF\_profile\_custom\_file, e\_failed\_to\_load\_set\_of\_bitmaps, e\_failed\_to\_set\_image\_overlay, e\_failed\_to\_set\_logo, e\_failed\_to\_play\_backwards, e\_failed\_to\_render\_file, e\_failed\_to\_renew\_recording\_file, e\_failed\_to\_set\_player\_speed\_ratio\_with\_audio, e\_failed\_to\_setup\_network\_streaming, e\_failed\_to\_start\_preview, e\_failed\_to\_start\_recording, e\_file\_in\_use, e\_file\_name\_not\_specified, e\_file\_not\_found, e\_get\_audio\_format, e\_get\_interface, e\_get\_video\_format, e\_graph\_error,

e\_graph\_cant\_run, e\_graph\_must\_be\_restarted, e\_hw\_deinterlace\_not\_supported, e\_incompatible\_options,  
 e\_index\_out\_of\_range, e\_invalid\_directory, e\_library\_not\_found, e\_load\_filter, e\_no\_audio\_input\_device,  
 e\_no\_device\_available, e\_no\_dialog\_for\_this\_compressor, e\_no\_stream\_control, e\_no\_tv\_tuner,  
 e\_no\_video\_device\_selected, e\_no\_video\_input\_device, e\_not\_allowed\_during\_network\_streaming,  
 e\_not\_assigned, e\_not\_multiplexed\_master, e\_not\_previewing, e\_not\_recording, e\_not\_reencoding,  
 e\_not\_streaming, e\_out\_of\_memory, e\_pause\_resume\_disabled, e\_pin\_not\_found,  
 e\_interface\_not\_assigned, e\_query\_config\_avi\_mux, e\_reencoding, e\_recording\_cannot\_pause,  
 e\_render\_audio\_stream, e\_render\_video\_stream, e\_must\_restart\_master, e\_recording\_on\_motion\_failed,  
 e\_sendtdv\_device\_index\_out\_of\_bound, e\_sendtdv\_deviceindex\_and\_videodevice\_have\_same\_value,  
 e\_sendtdv\_failed\_to\_bind\_dv\_device, e\_set\_filter\_graph, e\_set\_interleaving\_mode, e\_set\_master\_stream,  
 e\_set\_output\_compatibility\_index, e\_set\_output\_file\_name, e\_set\_format, e\_start\_preview\_first,  
 e\_stop\_player\_first, e\_stop\_preview\_first, e\_stop\_recording\_first, e\_stop\_reencoding\_first,  
 e\_storage\_path\_read\_only, e\_streaming\_type\_not\_specified, e\_third\_party\_filter\_already\_inserted,  
 e\_third\_party\_filter\_error, e\_trace\_log, e\_tv\_command\_not\_allowed\_during\_tv\_tuning,  
 e\_tuner\_input\_not\_selected, e\_TVideoGrabber\_Filter\_obsolete, e\_value\_out\_of\_range,  
 e\_video\_compressor\_not\_suitable, e\_window\_transparency\_failed,  
 i\_audio\_device\_associated\_to\_video\_device, i\_begin\_discovering\_device,  
 i\_binding\_device\_or\_compressor, i\_discovering\_device, i\_end\_discovering\_device,  
 i\_preallocated\_file\_size\_large\_enough, i\_preallocated\_file\_size\_changed, i\_preallocated\_file\_not\_suitable,  
 i\_streaming\_to\_publishing\_point, i\_third\_party\_filter\_inserted, i\_using\_ASF\_Profile,  
 i\_recording\_videosubtype, i\_ismpegstream, i\_new\_recording\_filename, i\_using\_property\_group,  
 w\_cannot\_connect\_thirdparty\_filter, w\_cannot\_connect\_thirdparty\_renderer,  
 w\_cannot\_instantiate\_thirdparty\_filter, w\_cannot\_route\_audio\_crossbar, w\_cannot\_use\_color\_key,  
 w\_command\_delayed, w\_does\_not\_apply\_to\_dv, w\_find\_audio\_device, w\_filter\_does\_not\_save\_properties,  
 w\_frame\_grabber\_requires\_CPU, w\_hold\_recording, w\_information, w\_not\_playing,  
 w\_player\_audio\_should\_be\_disabled, w\_recording\_cancelled\_by\_user,  
 w\_can\_pause\_and\_ASF\_incompatible, w\_set\_audio\_format, w\_storage\_path\_on\_network, w\_tv\_tuner,  
 w\_using\_nearest\_video\_size, w\_divx\_codec\_not\_installed,  
 w\_intervideo\_codec\_does\_not\_support\_debugger, w\_should\_install\_divx\_codec,  
 w\_device\_partially\_supported);

**enum** TLogType e\_add\_filter, e\_add\_source\_filter, e\_audio\_compressor\_not\_suitable,  
 e\_bind\_moniker\_to\_filter, e\_compressor\_possibly\_not\_suitable, e\_create\_instance,  
 e\_ddraw\_caps\_not\_suitable, e\_device\_in\_use\_in\_another\_graph, e\_disk\_full, e\_failed,  
 e\_failed\_to\_allocate\_recording\_file, e\_failed\_to\_bind\_codec, e\_failed\_to\_connect\_crossbar\_pin,  
 e\_failed\_to\_connect\_to\_server, e\_failed\_to\_create\_directory, e\_failed\_to\_create\_file,  
 e\_failed\_to\_create\_temp, e\_failed\_to\_bind\_frame\_grabber, e\_failed\_to\_load\_ASF\_profile,  
 e\_failed\_to\_load\_ASF\_profile\_custom\_file, e\_failed\_to\_load\_set\_of\_bitmaps,  
 e\_failed\_to\_set\_image\_overlay, e\_failed\_to\_set\_logo, e\_failed\_to\_play\_backwards, e\_failed\_to\_render\_file,  
 e\_failed\_to\_renew\_recording\_file, e\_failed\_to\_set\_player\_speed\_ratio\_with\_audio,  
 e\_failed\_to\_setup\_network\_streaming, e\_failed\_to\_start\_preview, e\_failed\_to\_start\_recording,  
 e\_file\_in\_use, e\_file\_name\_not\_specified, e\_file\_not\_found, e\_get\_audio\_format, e\_get\_interface,  
 e\_get\_video\_format, e\_graph\_error, e\_graph\_cant\_run, e\_graph\_must\_be\_restarted,  
 e\_hw\_deinterlace\_not\_supported, e\_incompatible\_options, e\_index\_out\_of\_range, e\_invalid\_directory,  
 e\_library\_not\_found, e\_load\_filter, e\_no\_audio\_input\_device, e\_no\_device\_available,  
 e\_no\_dialog\_for\_this\_compressor, e\_no\_stream\_control, e\_no\_tv\_tuner, e\_no\_video\_device\_selected,  
 e\_no\_video\_input\_device, e\_not\_allowed\_during\_network\_streaming, e\_not\_assigned,  
 e\_not\_multiplexed\_master, e\_not\_previewing, e\_not\_recording, e\_not\_reencoding, e\_not\_streaming,  
 e\_out\_of\_memory, e\_pause\_resume\_disabled, e\_pin\_not\_found, e\_interface\_not\_assigned,  
 e\_query\_config\_avi\_mux, e\_reencoding, e\_recording\_cannot\_pause, e\_render\_audio\_stream,  
 e\_render\_video\_stream, e\_must\_restart\_master, e\_recording\_on\_motion\_failed,  
 e\_sendtdv\_device\_index\_out\_of\_bound, e\_sendtdv\_deviceindex\_and\_videodevice\_have\_same\_value,  
 e\_sendtdv\_failed\_to\_bind\_dv\_device, e\_set\_filter\_graph, e\_set\_interleaving\_mode, e\_set\_master\_stream,  
 e\_set\_output\_compatibility\_index, e\_set\_output\_file\_name, e\_set\_format, e\_start\_preview\_first,  
 e\_stop\_player\_first, e\_stop\_preview\_first, e\_stop\_recording\_first, e\_stop\_reencoding\_first,  
 e\_storage\_path\_read\_only, e\_streaming\_type\_not\_specified, e\_third\_party\_filter\_already\_inserted,  
 e\_third\_party\_filter\_error, e\_trace\_log, e\_tv\_command\_not\_allowed\_during\_tv\_tuning,  
 e\_tuner\_input\_not\_selected, e\_TVideoGrabber\_Filter\_obsolete, e\_value\_out\_of\_range,  
 e\_video\_compressor\_not\_suitable, e\_window\_transparency\_failed,  
 i\_audio\_device\_associated\_to\_video\_device, i\_begin\_discovering\_device,  
 i\_binding\_device\_or\_compressor, i\_discovering\_device, i\_end\_discovering\_device,  
 i\_preallocated\_file\_size\_large\_enough, i\_preallocated\_file\_size\_changed, i\_preallocated\_file\_not\_suitable,

i\_streaming\_to\_publishing\_point, i\_third\_party\_filter\_inserted, i\_using\_ASF\_Profile,  
i\_recording\_videosubtype, i\_ismpegstream, i\_new\_recording\_filename, i\_using\_property\_group,  
w\_cannot\_connect\_thirdparty\_filter, w\_cannot\_connect\_thirdparty\_renderer,  
w\_cannot\_instantiate\_thirdparty\_filter, w\_cannot\_route\_audio\_crossbar, w\_cannot\_use\_color\_key,  
w\_command\_delayed, w\_does\_not\_apply\_to\_dv, w\_find\_audio\_device, w\_filter\_does\_not\_save\_properties,  
w\_frame\_grabber\_requires\_CPU, w\_hold\_recording, w\_information, w\_not\_playing,  
w\_player\_audio\_should\_be\_disabled, w\_recording\_cancelled\_by\_user,  
w\_can\_pause\_and\_ASF\_incompatible, w\_set\_audio\_format, w\_storage\_path\_on\_network, w\_tv\_tuner,  
w\_using\_nearest\_video\_size, w\_divx\_codec\_not\_installed,  
w\_intervideo\_codec\_does\_not\_support\_debugger, w\_should\_install\_divx\_codec,  
w\_device\_partially\_supported ;

## Description

**Possible constants returned by the [OnLog](#) event:**

e\_add\_filter  
e\_add\_source\_filter  
e\_audio\_compressor\_not\_suitable  
e\_bind\_moniker\_to\_filter  
e\_compressor\_possibly\_not\_suitable  
e\_create\_instance  
e\_ddraw\_caps\_not\_suitable  
e\_device\_in\_use\_in\_another\_graph  
e\_disk\_full  
e\_failed  
e\_failed\_to\_allocate\_recording\_file  
e\_failed\_to\_bind\_codec  
e\_failed\_to\_connect\_crossbar\_pin  
e\_failed\_to\_connect\_to\_server  
e\_failed\_to\_create\_directory  
e\_failed\_to\_create\_file  
e\_failed\_to\_create\_temp  
e\_failed\_to\_bind\_frame\_grabber  
e\_failed\_to\_load\_ASF\_profile  
e\_failed\_to\_load\_ASF\_profile\_custom\_file  
e\_failed\_to\_load\_set\_of\_bitmaps  
e\_failed\_to\_set\_image\_overlay  
e\_failed\_to\_set\_logo  
e\_failed\_to\_play\_backwards  
e\_failed\_to\_render\_file  
e\_failed\_to\_renew\_recording\_file  
e\_failed\_to\_set\_player\_speed\_ratio\_with\_audio  
e\_failed\_to\_setup\_network\_streaming  
e\_failed\_to\_start\_preview  
e\_failed\_to\_start\_recording  
e\_file\_in\_use  
e\_file\_name\_not\_specified  
e\_file\_not\_found  
e\_get\_audio\_format  
e\_get\_interface  
e\_get\_video\_format  
e\_graph\_error  
e\_graph\_cant\_run  
e\_graph\_must\_be\_restarted  
e\_hw\_deinterlace\_not\_supported  
e\_incompatible\_options  
e\_index\_out\_of\_range  
e\_invalid\_directory  
e\_library\_not\_found  
e\_load\_filter  
e\_no\_audio\_in\_device  
e\_no\_device\_available  
e\_no\_dialog  
e\_no\_stream\_control

e\_no\_tv\_tuner  
e\_no\_device\_selected  
e\_no\_video\_input\_device  
e\_not\_allowed\_during\_network\_streaming  
e\_not\_allowed\_with\_streaming\_URL  
e\_not\_assigned  
e\_not\_multiplexed\_master  
e\_not\_previewing  
e\_not\_recording  
e\_not\_reencoding  
e\_not\_streaming  
e\_out\_of\_memory  
e\_pause\_resume\_disabled  
e\_pin\_not\_found  
e\_interface\_not\_assigned  
e\_query\_config\_avi\_mux  
e\_reencoding  
e\_recording\_cannot\_pause  
e\_render\_audio\_stream  
e\_render\_video\_stream  
e\_must\_restart\_master  
e\_recording\_on\_motion\_failed  
e\_sendtdv\_device\_index\_out\_of\_bound  
e\_sendtdv\_deviceindex\_and\_videodevice\_have\_same\_value  
e\_sendtdv\_failed\_to\_bind\_dv\_device  
e\_set\_filter\_graph  
e\_set\_interleaving\_mode  
e\_set\_master\_stream  
e\_set\_output\_compatibility\_index  
e\_set\_output\_file\_name  
e\_set\_format  
e\_start\_preview\_first  
e\_stop\_player\_first  
e\_stop\_preview\_first  
e\_stop\_recording\_first  
e\_stop\_reencoding\_first  
e\_storage\_path\_read\_only  
e\_streaming\_type\_not\_specified  
e\_third\_party\_filter\_already\_inserted  
e\_third\_party\_filter\_error  
e\_trace\_log  
e\_tv\_command\_not\_allowed\_during\_tv\_tuning  
e\_tuner\_input\_not\_selected  
e\_TVideoGrabber\_Filter\_obsolete  
e\_value\_out\_of\_range  
e\_video\_compressor\_not\_suitable  
e\_window\_transparency\_failed  
e\_invalid\_size  
e\_invalid\_window\_handle  
e\_tuner\_mode\_not\_supported  
e\_publishing\_point\_connection\_failed  
e\_speaker\_control\_disabled  
i\_audio\_device\_associated\_to\_video\_device  
i\_begin\_discovering\_device  
i\_binding\_device\_or\_compressor  
i\_discovering\_device  
i\_end\_discovering\_device  
i\_preallocated\_file\_size\_large\_enough  
i\_preallocated\_file\_size\_changed  
i\_preallocated\_file\_not\_suitable  
i\_streaming\_to\_publishing\_point  
i\_third\_party\_filter\_inserted  
i\_using\_ASF\_Profile

i\_recording\_videosubtype  
i\_ismpegstream  
i\_new\_recording\_filename  
i\_using\_property\_group  
i\_streaming\_client\_connected  
i\_streaming\_client\_disconnected  
i\_refreshing\_preview  
i\_recording\_on\_motion  
i\_window\_found  
i\_limiting\_preview  
i\_codec\_recommended  
i\_tuner\_mode  
i\_DV\_date\_time\_discontinuity  
w\_cannot\_connect\_thirdparty\_filter  
w\_cannot\_connect\_thirdparty\_renderer  
w\_cannot\_instantiate\_thirdparty\_filter  
w\_cannot\_route\_crossbar  
w\_cannot\_use\_color\_key  
w\_command\_delayed  
w\_does\_not\_apply\_to\_dv  
w\_find\_audio\_device  
w\_filter\_does\_not\_save\_properties  
w\_frame\_grabber\_requires\_CPU  
w\_hold\_recording  
w\_information  
w\_not\_playing  
w\_player\_audio\_should\_be\_disabled  
w\_recording\_cancelled\_by\_user  
w\_can\_pause\_and\_ASF\_incompatible  
w\_set\_audio\_format  
w\_storage\_path\_on\_network  
w\_tv\_tuner  
w\_using\_nearest\_video\_size  
w\_divx\_codec\_not\_installed  
w\_codec\_does\_not\_support\_debugger  
w\_divx\_codec\_profile  
w\_device\_partially\_supported  
w\_excessive\_grid\_size  
w\_grid\_too\_large\_for\_dialog  
w\_operation\_may\_lock  
w\_audio\_streaming\_needs\_audiorecording\_property\_enabled  
w\_network\_streaming\_disabled  
w\_server\_lost\_next\_retry  
w\_overlay\_mixer\_not\_available  
w\_network\_streaming\_change\_requires\_application\_to\_be\_restarted  
w\_standard\_renderer\_recommended  
w\_window\_transparency\_and\_recording\_not\_recommended  
w\_clip\_not\_seekable  
w\_only\_WMV\_recording\_during\_network\_streaming  
w\_check\_analog\_video\_standard  
w\_recording\_timer\_set  
w\_stream\_time\_beyond\_script\_time  
w\_generate\_new\_file  
w\_hires\_timer\_not\_available  
w\_applies\_to\_the\_current\_recording\_method  
i\_leaving\_full\_screen\_mode  
i\_stream\_info  
i\_async\_url\_connection\_in\_progress  
i\_async\_url\_connection\_cancelled  
e\_obsolete  
i\_codec\_info  
i\_preview\_started  
i\_recording\_started

i\_reencoding\_started  
 i\_recording\_completed  
 i\_reencoding\_completed  
 i\_player\_opened  
 i\_inactive  
 i\_using\_stream\_index  
 e\_failed\_to\_start\_reencoding  
 e\_recording\_failed  
 e\_failed\_to\_open\_player

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Create High-Quality Help Documentation with a Help Authoring Tool](#)

---

## TMiscDeviceControl

### TMiscDeviceControl

[Prev](#)
[Next](#)

Type used by the [PutMiscDeviceData](#) and [GetMiscDeviceData](#) functions.

#### Description

Type used by the [PutMiscDeviceData](#) and [GetMiscDeviceData](#) functions.

Possible values:

**mdc\_GPIO:** used to set/retrieve the GPIO state

**mdc\_VPD:** used to set/retrieve the VPD (Virtual Private Data) state

**mdc\_VPD\_Data:** used to set/retrieve the VPD values.

---

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## TMouseButton

### TMouseButton

[Prev](#)
[Next](#)

TMouseButton type

#### Description

TMouseButton type

Possible values:

**mbLeft**

**mbRight**

**mbMiddle**

---

Created with the Standard Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

---

## TMPEGProgramSetting

### TMPEGProgramSetting

[Prev](#)
[Next](#)

TMPEGProgramSetting type

#### Declaration

```
TMPEGProgramSetting=(mps_Program_Number, mps_Program_PCR_PID, mps_VideoStream_PID,
mps_AudioStream_PID);
```

**enum** TMPEGProgramSetting mps\_Program\_Number, mps\_Program\_PCR\_PID, mps\_VideoStream\_PID, mps\_AudioStream\_PID

## Description

TMPEGProgramSetting type

mps\_Program\_Number  
mps\_Program\_PCR\_PID  
mps\_VideoStream\_PID  
mps\_AudioStream\_PID

---

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

---

## TMpegStreamType

### TMpegStreamType

[Prev](#)[Next](#)

Type of the Mpeg stream.

## Unit

[VidGrab](#)

## Declaration

TMpegStreamType=(mpst\_Default, mpst\_Program, mpst\_Program\_DVD, mpst\_Program\_DVD\_MC, mpst\_Program\_SVCD, mpst\_MPEG1, mpst\_MPEG1\_VCD);

**enum** TMpegStreamType mpst\_Default, mpst\_Program, mpst\_Program\_DVD, mpst\_Program\_DVD\_MC, mpst\_Program\_SVCD, mpst\_MPEG1, mpst\_MPEG1\_VCD ;

## Description

Type of the Mpeg stream.

## Possible values:

mpst\_Unknown  
mpst\_Program  
mpst\_Program\_DVD  
mpst\_Program\_DVD\_MC  
mpst\_Program\_SVCD  
mpst\_MPEG1  
mpst\_MPEG1\_VCD

---

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

---

## TMultiplexedRole

### TMultiplexedRole

[Prev](#)[Next](#)

Type of the [MultiplexedRole](#) property.

## Unit

[VidGrab](#)

## Declaration

TMultiplexedRole=(mr\_NotMultiplexed, mr\_MultiplexedMosaic4, mr\_MultiplexedMosaic16, mr\_MultiplexedMaster, mr\_MultiplexedSlave);

**enum** TMultiplexedRole mr\_NotMultiplexed, mr\_MultiplexedMosaic4, mr\_MultiplexedMosaic16, mr\_MultiplexedMaster, mr\_MultiplexedSlave ;

## Description

### Value Meaning



mr\_NotMultiplexed  
 mr\_MultiplexedMosaic4  
 mr\_MultiplexedMosaic16  
 mr\_MultiplexedMaster  
 mr\_MultiplexedSlave

#### See Also

[AssociateMultiplexedSlave](#) [EnableMultiplexedInput](#) [MultiplexedInputEmulation](#) [MultiplexedRole](#)  
[MultiplexedStabilizationDelay](#) [MultiplexedSwitchDelay](#) [UniqueID](#)

---

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)

---

## TNDIFormatType

### TNDIFormatType

[Prev](#)

[Next](#)

Type of NDI streaming format

#### Declaration

TNDIFormatType=(nft\_interleaved, nft\_progressive, nft\_field\_0, nft\_field\_1);

**enum** TNDIFormatType: **unsigned int** nft\_interleaved, nft\_progressive, nft\_field\_0, nft\_field\_1 ;

#### Description

##### Possible values:

nft\_interleaved  
 nft\_progressive  
 nft\_field\_0  
 nft\_field\_1

##### Default:

nft\_progressive

#### See Also

[NDIFormatType](#) [NDIName](#) [NetworkStreaming](#)

---

Created with the Standard Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

---

## TNetworkStreaming

### TNetworkStreaming

[Prev](#)

[Next](#)

Type of the [NetworkStreaming](#) property.

#### Unit

[VidGrab](#)

#### Declaration

TNetworkStreaming=(ns\_Disabled, ns\_ASFDirectNetworkStreaming, ns\_ASFStreamingToPublishingPoint);

**enum** TNetworkStreaming ns\_Disabled, ns\_ASFDirectNetworkStreaming,  
 ns\_ASFStreamingToPublishingPoint ;

#### Description

##### Value Meaning

ns\_Disabled network streaming disabled

ns\_ASFDirectNetworkStreaming direct network streaming from the platform running TVideoGrabber  
 ns\_ASFStreamingToPublishingPoint TVideoGrabber sends the streaming media to the Windows Media Server specified by [ASFMediaServerPublishingPoint](#).

#### See Also

[TNetworkStreamingType](#) [TOnClientConnection](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [News and information about help authoring tools and software](#)

## TNetworkStreamingType

### TNetworkStreamingType

[Prev](#)
[Next](#)

Type of the [NetworkStreamingType](#) property.

#### Declaration

```
TNetworkStreamingType=(nst_AudioVideoStreaming, nst_VideoStreaming, nst_AudioStreaming);
```

```
enum TNetworkStreamingType _AudioVideoStreaming, nst_VideoStreaming, nst_AudioStreaming
```

#### Description

##### Values:

nst\_AudioVideoStreaming : audio + video streaming  
 nst\_VideoStreaming : video streaming only  
 nst\_AudioStreaming : audio streaming only

#### See Also

[TOnClientConnection](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Make Your PDFs More Secure with Encryption and Password Protection](#)

## TNotificationMethod

### TNotificationMethod

[Prev](#)
[Next](#)

Type of [NotificationMethod](#) property.

#### Unit

[VidGrab](#)

**Declaration**

TNotificationMethod=(nm\_Timer, nm\_Thread);

**enum** TNotificationMethod nm\_Timer, nm\_Thread ;

**Description****Value Meaning**

nm\_Timer notification by timer

nm\_Thread notification by thread

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

**TOnAudioBufferNegotiation****TOnAudioBufferNegotiation**
[Prev](#)
[Next](#)

TOnAudioBufferNegotiation

**Declaration**

TOnAudioBufferNegotiation=procedure (Sender: TObject; nChannels: LongInt; nSamplesPerSec: LongInt; BitsPerSample: LongInt; var ProposedBufferSize: LongInt);

**Description**

Type of the [OnAudioBufferNegotiation](#) event

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

**TOnAudioPeak****TOnAudioPeak**
[Prev](#)
[Next](#)

Type of the [OnAudioPeak](#) event.

**Declaration**

TOnAudioPeak=**procedure** (Sender: TObject; Left\_Percent: Double; Left\_DB: Double; Right\_Percent: Double; Right\_DB: Double) **of object**;

**typedef void \_\_fastcall**(\_\_closure \*TOnAudioPeak)(System::TObject \*Sender, double Left\_Percent, Left\_DB, Right\_Percent, Right\_DB);

**Description**

Type of the [OnAudioPeak](#) event.

Created with the Standard Edition of HelpNDoc: [Easy CHM and documentation editor](#)

**TOnAuthenticationNeeded****TOnAuthenticationNeeded**
[Prev](#)
[Next](#)

Type of the [OnAuthenticationNeeded](#) event.

**Declaration**

TOnAuthenticationNeeded=**procedure** (Sender: TObject; Id: LongInt; Realm: **String**; Server: **string**; var Username: **string**; var Password: **string**);

```
typedef void __fastcall (__closure *TOnAuthenticationNeeded)(System::TObject* Sender, int Id, wchar_t *Realm, wchar_t *Server, wchar_t *&Username, wchar_t *&Password);
```

### Description

Type of the [OnAuthenticationNeeded](#) event.

Created with the Standard Edition of HelpNDoc: [Full-featured Documentation generator](#)

## TOnAVIDurationUpdated

### TOnAVIDurationUpdated

[Prev](#)
[Next](#)

Type of the [OnEndOfAVIRecording](#) event.

### Declaration

TOnAVIDurationUpdated=**procedure**(Sender: TObject; FileName: **string**; FrameCount: LargeInteger; **var** FrameRate: Double; **var** Duration: LargeInteger) **of object**;

```
typedef void __fastcall(__closure *TOnAVIDurationUpdated)(System::TObject *Sender, wchar_t *FileName, __int64 FrameCount, double &FrameRate, __int64 &Duration);
```

### Description

Type of the [OnEndOfAVIRecording](#) event.

Created with the Standard Edition of HelpNDoc: [Effortlessly Convert Your Word Doc to an eBook: A Step-by-Step Guide](#)

## TOnClientConnection

### TOnClientConnection

[Prev](#)
[Next](#)

Type of the [OnClientConnection](#) event.

### Declaration

TOnClientConnection=**procedure** (Sender: TObject; Connected: Boolean; ClientInfo: **String**)**of object**;

```
typedef void __fastcall (__closure *TOnClientConnection)(System::TObject* Sender, bool Connected, wchar_t *ClientInfo);
```

### Description

Type of the [OnClientConnection](#) event.

### See Also

[TNetworkStreamingType](#) [TNetworkStreaming](#) [TOnDirectNetworkStreamingHostUrl](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## TOnColorKeyChange

### TOnColorKeyChange

[Prev](#)
[Next](#)

The type of the [OnColorKeyChange](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnColorKeyChange=**procedure**(Sender: TObject; ColorKey: TColor) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnColorKeyChange)(System::TObject \*Sender, Graphics::TColor ColorKey);

#### Description

The type of the [OnColorKeyChange](#) event.

**ColorKey**: new color key used for window transparency.

#### See Also

[ColorKeyEnabled](#) [OnColorKeyChange](#)

Created with the Standard Edition of HelpNDoc: [Free CHM Help documentation generator](#)

## TOnCreatePreallocatedFileCompleted

### TOnCreatePreallocatedFileCompleted

[Prev](#)
[Next](#)

Type of the [OnCreatePreallocatedFileCompleted](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnCreatePreallocatedFileCompleted=**procedure**(Sender: TObject; FileName: **String**; Success: Boolean) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnCreatePreallocatedFileCompleted)(System::TObject \*Sender, wchar\_t \*FileName, **bool** Success);

#### Description

Type of the [OnCreatePreallocatedFileCompleted](#) event.

#### See Also

[OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileProgress](#) [OnCreatePreallocFileStarted](#)

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## TOnDeviceArrivalOrRemoval

### TOnDeviceArrivalOrRemoval

[Prev](#)
[Next](#)

The type of the [OnDeviceArrivalOrRemoval](#) event.

**Unit**[VidGrab](#)**Declaration**

TOnDeviceArrivalOrRemoval=**procedure**(Sender: TObject; IsDeviceArrival: Boolean; IsVideoDevice: Boolean; DeviceName: **string**; DeviceIndex: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnDeviceArrivalOrRemoval)(System::TObject *Sender, bool IsDeviceArrival, bool IsVideoDevice, wchar_t *DeviceName, int DeviceIndex);
```

**Description**

The type of the [OnDeviceArrivalOrRemoval](#) event.

**IsDeviceArrival**: true if the device has been connected, false if the device has been disconnected

**IsVideoDevice**: true if the device is a video capture device, false if the device is an audio capture device.

**DeviceName**: name of the related capture device.

**DeviceIndex**: index of the capture device in the [VideoDevices](#) list or [AudioDevices](#) list, according to IsVideoDevice.

**See Also**

[OnDeviceArrivalOrRemoval](#) [OnDeviceLost](#) [RefreshDevicesAndCompressorsLists](#)

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

**TOnDirectNetworkStreamingHostUrl****TOnDirectNetworkStreamingHostUrl**[Prev](#)[Next](#)

Type of the [OnDirectNetworkStreamingHostUrl](#) event.

**Unit**[VidGrab](#)**Declaration**

TOnDirectNetworkStreamingHostUrl=**procedure**(Sender: TObject; HostUrl: **String**; HostName: **String**; HostPort: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnDirectNetworkStreamingHostUrl)(System::TObject *Sender, wchar_t *HostUrl, wchar_t *HostName, int HostPort);
```

**Description**

Type of the [OnDirectNetworkStreamingHostUrl](#) event.

**HostUrl**: the full qualified URL on which users must connect to get live streaming on the current platform

**HostName**: host name on the current platform

**HostPort**: streaming port on the current platform

**See Also**

[TNetworkStreamingType](#) [TOnClientConnection](#) [TNetworkStreaming](#) [ASFAudioBitRate](#) [ASFAudioChannels](#) [ASFDeinterlaceMode](#) [ASFFixedFrameRate](#) [ASFMediaServerPublishingPoint](#) [ASFMediaServerRemovePublishingPointAfterDisconnect](#) [ASFMediaServerTemplatePublishingPoint](#) [ASFNetworkMaxUsers](#) [ASFNetworkPort](#) [ASFProfile](#) [ASFProfileFromCustomFile](#) [ASFProfiles](#) [ASFProfilesCount](#) [ASFProfileVersion](#) [ASFVideoBitRate](#) [ASFVideoFrameRate](#) [ASFVideoHeight](#) [ASFVideoMaxKeyFrameSpacing](#) [ASFVideoQuality](#) [ASFVideoWidth](#) [NetworkStreaming](#) [NetworkStreamingType](#) [OnAuthenticationNeeded](#) [OnClientConnection](#) [OnDirectNetworkStreamingHostUrl](#) [ShowDialog](#) [StartAudioRendering](#) [StreamingURL](#)

Created with the Standard Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

## TOnDragDropFiles

### TOnDragDropFiles

[Prev](#)
[Next](#)

TOnDragDropFiles

#### Declaration

TOnDragDropFiles=**procedure** (Sender: Pointer; FileCount: LongInt; FirstFile: **string**; FilesList: **string**);

#### Description

Type of the [OnDragDropFiles](#) event

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with HelpNDoc's CHM Help File Creation Features](#)

## TOnDVCommandCompleted

### TOnDVCommandCompleted

[Prev](#)
[Next](#)

Type of the [OnDVCommandCompleted](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnDVCommandCompleted=**procedure**(Sender: TObject; NewStateValue: LongInt; NewStateLabel: **String**)**of object**;

```
typedef void __fastcall(__closure *TOnDVCommandCompleted)(System::TObject *Sender, int
NewStateValue, wchar_t *NewStateLabel);
```

#### Description

Type of the [OnDVCommandCompleted](#) event.

#### See Also

[TDVSize](#) [TDVVideoFormat](#) [TDVVideoStandard](#) [DVDDateTimeEnabled](#) [DVDDiscontinuityMinimumInterval](#) [DVRecordingInNativeFormatSeparatesStreams](#) [DVReduceFrameRate](#) [DVRgb219](#) [DVTimeCodeEnabled](#) [IsDigitalVideoIn](#) [IsDVCommandAvailable](#) [IsTimeCodeReaderAvailable](#) [OnDVCommandCompleted](#) [SendDVCommand](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## TOnDVDDiscontinuity

### TOnDVDDiscontinuity

[Prev](#)
[Next](#)

Type of the [OnDVDDiscontinuity](#) event.

#### Declaration

TOnDVDDiscontinuity=**procedure**(Sender: TObject; var DeliverNewFrame: Boolean)**of object**;

```
typedef void __fastcall(__closure *TOnDVDDiscontinuity)(System::TObject *Sender, bool
*DeliverNewFrame);
```

#### Description



Type of the [OnDVDDiscontinuity](#) event.

Created with the Standard Edition of HelpNDoc: [Easily create iPhone documentation](#)

## TOnEnumerateWindows

### TOnEnumerateWindows

[Prev](#)
[Next](#)

Type of the [OnEnumerateWindows](#) event.

#### Declaration

TOnEnumerateWindows=**procedure** (Sender: TObject; WindowHandle: LongInt; WindowName: String; WindowClass: String) **of object**;

```
typedef void __fastcall (__closure *TOnEnumerateWindows)(System::TObject* Sender, int WindowHandle, wchar_t *WindowName, wchar_t *WindowClass);
```

#### Description

Type of the [OnEnumerateWindows](#) event.

#### See Also

[EnumerateWindows](#) [ScreenRecordingLayeredWindows](#) [ScreenRecordingMonitor](#) [ScreenRecordingNonVisibleWindows](#) [ScreenRecordingUsingCoordinates](#) [ScreenRecordingWithCursor](#) [SetWindowRecordingByHandle](#) [SetWindowRecordingByName](#)

Created with the Standard Edition of HelpNDoc: [Transform Your Documentation Process with HelpNDoc's Project Analyzer](#)

## TOnFileNotification

### TOnFileNotification

[Prev](#)
[Next](#)

Type of the [OnRecordingStarted](#) and the [OnCreatePreallocFileStarted](#) events.

#### Unit

[VidGrab](#)

#### Declaration

TOnFileNotification=**procedure**(Sender: TObject; FileName: **String**)**of object**;

```
typedef void __fastcall(__closure *TOnFileNotification)(System::TObject *Sender, wchar_t *FileName);
```

#### Description

Type of the [OnRecordingStarted](#) and the [OnCreatePreallocFileStarted](#) events.

Created with the Standard Edition of HelpNDoc: [Easily create Help documents](#)

## TOnFilterSelected

### TOnFilterSelected

[Prev](#)
[Next](#)

Type of the OnFilterSelected event

#### Declaration

TOnFilterSelected=**procedure** (Sender: TObject; FilterName: **string**; **var** RejectFilter: Boolean)

```
typedef void __fastcall(__closure *TOnFilterSelected)(System::TObject *Sender, wchar_t *FilterName, bool &RejectFilter)
```

### Description

Type of the OnFilterSelected event

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## TOnFrameCaptureCompleted

### TOnFrameCaptureCompleted

[Prev](#)

[Next](#)

Type of the [OnFrameCaptureCompleted](#) event.

### Unit

[VidGrab](#)

### Declaration

TOnFrameCaptureCompleted=**procedure**(Sender: TObject; FrameBitmap: Graphics.TBitmap; FrameBitmap: Graphics.TBitmap; BitmapWidth: LongInt; BitmapHeight: LongInt; FrameNumber: LongWord; FrameTime: LargeInteger; DestType: TFrameCaptureDest; FileName: **string**; Success: Boolean; FrameId: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnFrameCaptureCompleted)(System::TObject *Sender, Graphics_TBitmap *FrameBitmap, int BitmapWidth, int BitmapHeight, unsigned FrameNumber, __int64 FrameTime, TFrameCaptureDest DestType, wchar_t *FileName, bool Success, int FrameId);
```

### Description

Type of the [OnFrameCaptureCompleted](#) event.

The values returned are described in the [OnFrameCaptureCompleted](#) event.

### See Also

[TAutoFileName](#) [TFrameCaptureDest](#) [BurstCount](#) [BurstInterval](#) [BurstMode](#) [BurstType](#) [CaptureFrameSyncTo](#) [CaptureFrameTo](#) [FrameCaptureHeight](#) [FrameCaptureWidth](#) [FrameCaptureWithoutOverlay](#) [FrameCaptureZoomSize](#) [GetFrameInfoString](#) [GetLastFrameAsHBITMAP](#) [GetLastFrameAsTBitmap](#) [GetLastFrameBitmapBits](#) [GetLastFrameBitmapBits2](#) [GetLastFrameWaitTimeoutMs](#) [JPEGPerformance](#) [JPEGProgressiveDisplay](#) [JPEGQuality](#) [Last](#) [BurstFrameCapture](#) [FileName](#) [Last](#) [CaptureFrameTo](#) [FileName](#) [OnDiskFull](#) [OnFrameBitmap](#) [SetFrameCaptureBounds](#) [StoragePath](#) [WebcamStillCaptureButton](#)

Created with the Standard Edition of HelpNDoc: [Keep Your PDFs Safe from Unauthorized Access with These Security Measures](#)

## TOnFrameOverlayUsingDC

### TOnFrameOverlayUsingDC

[Prev](#)

[Next](#)

Type of the [OnFrameOverlayUsingDC](#) event.

### Unit

[VidGrab](#)

### Declaration

TOnFrameOverlayUsingDC=**procedure**(Sender: TObject; Dc: HDC; FrameNumber: LongWord; FrameTime: LargeInteger; FrameId: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnFrameOverlayUsingDC)(System::TObject *Sender, HDC Dc,
unsigned FrameNumber, __int64 FrameTime, int FrameId);
```

### Description

Type of the [OnFrameOverlayUsingDC](#) event.

**Dc** is the current device context of the current video frame.

**FrameId** must be passed to [GetFrameInfo](#) to retrieve information about the current frame.

Use the **Dc** device context to draw objects over video frames by using GDI functions.

### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDIB](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#)  
[DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchron](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

Created with the Standard Edition of HelpNDoc: [Produce Kindle eBooks easily](#)

## TOnFrameOverlayUsingDIB

### TOnFrameOverlayUsingDIB

[Prev](#)
[Next](#)

Type of the OnFrameOverlayUsingDIB event.

### Unit

[VidGrab](#)

### Declaration

TOnFrameOverlayUsingDIB=**procedure**(Sender: TObject; FrameBitmapHandle: HBITMAP; FrameNumber: LongWord; FrameTime: LargeInteger; FrameId: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnFrameOverlayUsingDIB)(System::TObject *Sender, HBITMAP
FrameBitmapHandle, unsigned FrameNumber, __int64 FrameTime, int FrameId);
```

### Description

Type of the [OnFrameOverlayUsingDIB](#) event.

**FrameBitmapHandle:** handle of the bitmap of the current video frame

**FrameId:** must be passed to [GetFrameInfo](#) to retrieve information about the current frame.

Use [GetObject](#) to retrieve the DIB section information. E.g.:

```
procedure TfrmMainForm.VideoGrabberFrameOverlayUsingDIB(Sender: TObject;
  FrameBitmapHandle: HBITMAP; FrameNumber: Cardinal; FrameTime: Int64;
  FrameId: Integer);
var
  DibSection: TDibSection;
  i: integer;
  SkipLine: boolean;
begin
  if GetObject (FrameBitmapHandle, sizeof (TDibSection), @DibSection) = 0 then Exit;
  SkipLine := false;
  for i := 0 to DibSection.dsBm.bmHeight - 1 do begin
    if SkipLine then begin
      ZeroMemory (Pointer (Integer(DibSection.dsBm.bmBits) + (i * DibSection.dsBm.bmPitch)),
        sizeof (DibSection.dsBm.bmBits));
    end;
    SkipLine := not SkipLine;
  end;
end;
```

#### See Also

[TCardinalDirection](#) [TOnFrameOverlayUsingDC](#) [TTextOverlayAlign](#) [AdjustOverlayAspectRatio](#)  
[DrawBitmapOverFrame](#) [GetFrameInfoString](#) [ImageOverlay](#) [StretchToVideoSize](#) [MouseWheelEventEnabled](#)  
[OnFrameBitmap](#) [OnFrameBitmapEventSynchronise](#) [OnFrameOverlayUsingDC](#) [OnFrameOverlayUsingDIB](#)  
[OnFrameOverlayUsingVIDEOHDR](#) [OnMouseDown](#) [OnMouseMove](#) [OnMouseUp](#) [OnMouseWheel](#)  
[OverlayAfterTransform](#) [RefreshPlayerOverlays](#) [SetImageOverlay](#) [AlphaBlend](#)  
[SetImageOverlay](#) [AlphaBlendValue](#) [SetImageOverlay](#) [ChromaKey](#)  
[SetImageOverlay](#) [ChromaKeyLeewayPercent](#) [SetImageOverlay](#) [ChromaKeyRGBColor](#)  
[SetImageOverlay](#) [Enabled](#) [SetImageOverlay](#) [Height](#) [SetImageOverlay](#) [LeftLocation](#)  
[SetImageOverlay](#) [RotationAngle](#) [SetImageOverlay](#) [StretchToVideoSize](#) [SetImageOverlay](#) [TargetDisplay](#)  
[SetImageOverlay](#) [TopLocation](#) [SetImageOverlay](#) [Transparent](#) [SetImageOverlay](#) [TransparentColorValue](#)  
[SetImageOverlay](#) [UseTransparentColor](#) [SetImageOverlay](#) [Width](#) [SetImageOverlayFromBMPFile](#)  
[SetImageOverlayFromHBitmap](#) [SetImageOverlayFromHBitmap2](#) [SetImageOverlayFromImageFile](#)  
[SetImageOverlayFromImageFile2](#) [SetImageOverlayFromJPEGFile](#) [SetImageOverlayFromTBitmap](#)  
[SetImageOverlayFromTBitmap2](#) [SetImageOverlayFromTImage](#) [SetImageOverlayFromTImage2](#)  
[SetTextOverlay](#) [Align](#) [SetTextOverlay](#) [BkColor](#) [SetTextOverlay](#) [CustomVar](#) [SetTextOverlay](#) [Enabled](#)  
[SetTextOverlay](#) [Font](#) [SetTextOverlay](#) [FontColor](#) [SetTextOverlay](#) [GradientColor](#)  
[SetTextOverlay](#) [GradientMode](#) [SetTextOverlay](#) [HighResFont](#) [SetTextOverlay](#) [Left](#) [SetTextOverlay](#) [Right](#)  
[SetTextOverlay](#) [Scrolling](#) [SetTextOverlay](#) [ScrollingSpeed](#) [SetTextOverlay](#) [Shadow](#)  
[SetTextOverlay](#) [ShadowColor](#) [SetTextOverlay](#) [ShadowDirection](#) [SetTextOverlay](#) [String](#)  
[SetTextOverlay](#) [TargetDisplay](#) [SetTextOverlay](#) [Top](#) [SetTextOverlay](#) [Transparent](#) [ShapeOverlay](#)  
[ShapeOverlayEnabled](#) [ShapeOverlayList](#) [TextOverlay](#) [Align](#) [TextOverlay](#) [BkColor](#)  
[TextOverlay](#) [CreateCustomFont](#) [TextOverlay](#) [CreateCustomFont2](#) [TextOverlay](#) [Enabled](#) [TextOverlay](#) [Font](#)  
[TextOverlay](#) [FontColor](#) [TextOverlay](#) [Left](#) [TextOverlay](#) [Right](#) [TextOverlay](#) [Scrolling](#)  
[TextOverlay](#) [ScrollingSpeed](#) [TextOverlay](#) [Selector](#) [TextOverlay](#) [Shadow](#) [TextOverlay](#) [ShadowColor](#)  
[TextOverlay](#) [ShadowDirection](#) [TextOverlay](#) [String](#) [TextOverlay](#) [Top](#) [TextOverlay](#) [Transparent](#)  
[TranslateMouseCoordinates](#)

## TOnFrameProgress

### TOnFrameProgress

[Prev](#)
[Next](#)

The type of the [OnFrameProgress](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnFrameProgress=**procedure**(Sender: TObject; FrameNumber: LongWord; FrameTime: LargeInteger; FrameId: LongInt) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnFrameProgress)(System::TObject \*Sender, **unsigned** FrameNumber, \_\_int64 FrameTime, **int** FrameId);

#### Description

The type of the [OnFrameProgress](#) event.

**FrameId** must be passed to [GetFrameInfo](#) to retrieve information about the current frame.

*Look at the MainDemo project for sample code.*

#### See Also

[TOnResizeVideo](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Easy EPub and documentation editor](#)

## TOnLog

### TOnLog

[Prev](#)
[Next](#)

The type of the [OnLog](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnLog=**procedure**(Sender: TObject; LogType: TLogType; Severity: **string**; InfoMsg: **string**)**of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnLog)(System::TObject \*Sender, TLogType LogType, wchar\_t \*Severity, wchar\_t \*InfoMsg);

#### Description

The type of the [OnLog](#) event.

**Severity** returns "[INFO]", "[WARNING]" or "[ERROR]".

**LogType** returns a [TLogType](#) constant.

**InfoMsg** returns a comment about this log.

Note: it is possible to retrieve the string value of the LogType parameter by invoking [GetLogString](#) (LogType).

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## TOnMotionDetected

### TOnMotionDetected

[Prev](#)
[Next](#)

Type of the [OnMotionDetected](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnMotionDetected=**procedure**(Sender: TObject; GlobalMotionRatio: Double; MaxMotionCellX: LongInt; MaxMotionCellY: LongInt; FrameBitmap: Graphics.TBitmap; FrameBitmap: Graphics.TBitmap; FrameNumber: LongWord; FrameTime: LargeInteger; Frameld: LongInt; **var** CaptureFrame: Boolean) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnMotionDetected)(System::TObject \*Sender, **double** GlobalMotionRatio, **int** MaxMotionCellX, **int** MaxMotionCellY, Graphics.TBitmap \*FrameBitmap, **unsigned** FrameNumber, \_\_int64 FrameTime, **int** Frameld, **bool** &CaptureFrame);

#### Description

Type of the [OnMotionDetected](#) event.

**GlobalMotionRatio**: motion ratio of the whole video frame, compared to the previous one

**FrameData**: returns the bitmap of the video frame, the frame number and the frame time

**CaptureFrame**: set this parameter to TRUE if you want to capture this frame. In this case it will be returned by the OnFrameCaptureCompleted event.

#### See Also

[Color / Greyscale Grid structure / grid sensitivity Motion ratio Recording only when motion is detected](#)  
[Video noise TOnMotionNotDetected MotionDetector CellMotionRatio MotionDetector CompareBlue MotionDetector CompareGreen MotionDetector CompareRed MotionDetector Enabled MotionDetector EnumGridDialogControls MotionDetector Get2DTextGrid MotionDetector Get2DTextMotion MotionDetector GetCellLocation MotionDetector GetCellSensitivity MotionDetector GetCellSize MotionDetector GloballyIncOrDecSensitivity MotionDetector GlobalMotionRatio MotionDetector GreyScale MotionDetector Grid MotionDetector GridXCount MotionDetector GridYCount MotionDetector IsGridValid MotionDetector MaxDetectionsPerSecond MotionDetector ReduceCPULoad MotionDetector ReduceVideoNoise MotionDetector Reset MotionDetector ResetGlobalSensitivity MotionDetector SetCellSensitivity MotionDetector SetGridSize MotionDetector ShowGridDialog MotionDetector Triggered MotionDetector UseThisReferenceSample OnBacktimedFramesCountReached OnMotionDetected OnMotionNotDetected RecordingOnMotion Enabled RecordingOnMotion MotionThreshold RecordingOnMotion NoMotionPauseDelayMs](#)

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

## TOnMotionNotDetected

### TOnMotionNotDetected

[Prev](#)
[Next](#)

Type of the [OnMotionNotDetected](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnMotionNotDetected=**procedure**(Sender: TObject; FrameBitmap: Graphics.TBitmap; FrameBitmap: Graphics.TBitmap; FrameNumber: LongWord; FrameTime: LargeInteger; Frameld: LongInt; **var** CaptureFrame: Boolean) **of object**;

```
typedef void __fastcall(__closure *TOnMotionNotDetected)(System::TObject *Sender, Graphics_TBitmap *FrameBitmap, unsigned FrameNumber, __int64 FrameTime, int FrameId, bool &CaptureFrame);
```

### Description

Type of the [OnMotionNotDetected](#) event.

**FrameData:** returns the bitmap of the video frame, the frame number and the frame time

**CaptureFrame:** set this parameter to TRUE if you want to capture this frame. In this case it will be returned by the OnFrameCaptureCompleted event.

### See Also

[Color / Greyscale Grid structure / grid sensitivity](#) [Motion ratio](#) [Recording only when motion is detected](#) [Video noise](#) [TOnMotionDetected](#) [MotionDetector](#) [CellMotionRatio](#) [MotionDetector](#) [CompareBlue](#) [MotionDetector](#) [CompareGreen](#) [MotionDetector](#) [CompareRed](#) [MotionDetector](#) [Enabled](#) [MotionDetector](#) [EnumGridDialogControls](#) [MotionDetector](#) [Get2DTextGrid](#) [MotionDetector](#) [Get2DTextMotion](#) [MotionDetector](#) [GetCellLocation](#) [MotionDetector](#) [GetCellSensitivity](#) [MotionDetector](#) [GetCellSize](#) [MotionDetector](#) [GloballyIncOrDecSensitivity](#) [MotionDetector](#) [GlobalMotionRatio](#) [MotionDetector](#) [GreyScale](#) [MotionDetector](#) [Grid](#) [MotionDetector](#) [GridXCount](#) [MotionDetector](#) [GridYCount](#) [MotionDetector](#) [IsGridValid](#) [MotionDetector](#) [MaxDetectionsPerSecond](#) [MotionDetector](#) [ReduceCPULoad](#) [MotionDetector](#) [ReduceVideoNoise](#) [MotionDetector](#) [Reset](#) [MotionDetector](#) [ResetGlobalSensitivity](#) [MotionDetector](#) [SetCellSensitivity](#) [MotionDetector](#) [SetGridSize](#) [MotionDetector](#) [ShowGridDialog](#) [MotionDetector](#) [Triggered](#) [MotionDetector](#) [UseThisReferenceSample](#) [OnBacktimedFramesCountReached](#) [OnMotionDetected](#) [OnMotionNotDetected](#) [RecordingOnMotion](#) [Enabled](#) [RecordingOnMotion](#) [MotionThreshold](#) [RecordingOnMotion](#) [NoMotionPauseDelayMs](#)

---

Created with the Standard Edition of HelpNDoc: [Converting Word Docs to eBooks Made Easy with HelpNDoc](#)

---

## TOnMouseWheel

### TOnMouseWheel

[Prev](#)
[Next](#)

Type of the [OnMouseWheel](#) event

### Declaration

TOnMouseWheel=**procedure** (Sender: TObject; VideoWindow: LongInt; DirectionUp: Boolean; X: LongInt; Y: LongInt) **of object**;

```
typedef void __fastcall (__closure *TOnMouseWheel)(System::TObject* Sender, int VideoWindow, bool DirectionUp, int X, int Y);
```

### Description

Type of the [OnMouseWheel](#) event.

The DirectionUp parameter returns:

- true when the mouse wheel goes up,
- false when the mouse wheel goes down.

---

Created with the Standard Edition of HelpNDoc: [Transform Your Help Documentation Process with a Help Authoring Tool](#)

---

## TOnONVIFDiscoveryCompletedNotification

### TOnONVIFDiscoveryCompletedNotification

[Prev](#)
[Next](#)

Type of the [OnONVIFDiscoveryCompletedNotification](#) event



## Declaration

TOnONVIFDiscoveryCompletedNotification=**procedure** (Sender: TObject; DiscoveryCallbackStatus: TDiscoveryCallbackStatus; CameraCount: LongInt) **of object**;

## Description

**DiscoveryCallbackStatus: TDiscoveryCallbackStatus:**

specifying whether the event occurs for a new camera discovered, or for the completion of the Multicast or IP range discovery

**CameraCount: LongInt:**

number of cameras currently discovered

## See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TONVIFDeviceInfo](#) [Video stream of IP camera with audio of PC microphone or other audio capture device](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [IPCameraURL](#) [IsURLResponding](#) [IsURLVideoStreamAvailable](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras](#) [Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapshot](#) [OpenURLAsync](#) [OpenURLAsyncStatus](#) [SendIPCameraCommand](#) [SetIPCameraSetting](#)

Created with the Standard Edition of HelpNDoc: [Write eBooks for the Kindle](#)

## TOnOpenURLAsyncStatusChanged

### TOnOpenURLAsyncStatusChanged

[Prev](#)

[Next](#)

Type of the [OnOpenURLAsyncStatusChanged](#) event

## Declaration

TOnOpenURLAsyncStatusChanged=**procedure** (Sender: TObject; OldStatus: TOpenURLAsyncStatus; NewStatus: TOpenURLAsyncStatus; TargetState: TCurrentState) **of object**;

**typedef void \_\_fastcall (\_\_closure \*TOnOpenURLAsyncStatusChanged)(System::TObject\* Sender, TOpenURLAsyncStatus OldStatus, TOpenURLAsyncStatus NewStatus, TCurrentState TargetState);**

Created with the Standard Edition of HelpNDoc: [Effortlessly Support Your Windows Applications with HelpNDoc's CHM Generation](#)

## TOnPlayerBufferingData

### TOnPlayerBufferingData

[Prev](#)

[Next](#)

Type of the [OnPlayerBufferingData](#) event.

## Unit

[VidGrab](#)

## Declaration

TOnPlayerBufferingData=**procedure**(Sender: TObject; StartingToBuffer: Boolean) **of object**;

**typedef void \_\_fastcall(\_\_closure \*TOnPlayerBufferingData)(System::TObject \*Sender, bool StartingToBuffer);**

## Description

Type of the [OnPlayerBufferingData](#) event.

**StartingToBuffer** returns true when the player starts buffering data.

**StartingToBuffer** returns false when the player ends buffering data.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TPlayerState](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer](#)

## TOnPlayerStateChanged

### TOnPlayerStateChanged

[Prev](#)
[Next](#)

Type of the [OnPlayerStateChanged](#) event.

#### Declaration

TOnPlayerStateChanged=**procedure** (Sender: TObject; OldPlayerState: TPlayerState; NewPlayerState: TPlayerState) of **object**

```
typedef void __fastcall (__closure *TOnPlayerStateChanged)(System::TObject* Sender,
TPlayerState OldPlayerState, TPlayerState NewPlayerState);
```

#### Description

Type of the [OnPlayerStateChanged](#) event.

#### See Also

[Player features](#) [TPlayerState](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#) [PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchronise](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## TOnProgress

### TOnProgress

[Prev](#)
[Next](#)

Type of the [OnCopyPreallocDataProgress](#) and [OnReencodingProgress](#) events.

#### Unit

[VidGrab](#)

#### Declaration

TOnProgress=**procedure**(Sender: TObject; Percent: LongInt; Position: LargeInteger; Duration: LargeInteger) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnProgress)(System::TObject \*Sender, **int** Percent, \_\_int64 Position, \_\_int64 Duration);

#### Description

Type of the [OnCopyPreallocDataProgress](#) and [OnReencodingProgress](#) events.

Created with the Standard Edition of HelpNDoc: [Create iPhone web-based documentation](#)

## TOnProgressCommented

### TOnProgressCommented

[Prev](#)

[Next](#)

Type of the [OnBitmapsLoadingProgress](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnProgressCommented=**procedure**(Sender: TObject; Percent: LongInt; Position: LargeInteger; Duration: LargeInteger; Comment: **String**)**of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnProgressCommented)(System::TObject \*Sender, **int** Percent, \_\_int64 Position, \_\_int64 Duration, wchar\_t \*Comment);

#### Description

Type of the [OnBitmapsLoadingProgress](#) event.

Created with the Standard Edition of HelpNDoc: [Maximize Your Documentation Capabilities with HelpNDoc's Project Analyzer](#)

## TOnRawSample

### TOnRawSample

[Prev](#)

[Next](#)

Type of the [OnRawVideoSample](#) and [OnRawAudioSample](#) events.

#### Unit

[VidGrab](#)

#### Declaration

TOnRawSample=**procedure**(Sender: TObject; pSampleBuffer: PByte; SampleBufferSize: LongInt; SampleDataLength: LongInt; FormatType: TFormatType; pFormat: Pointer; pBitmapInfoHeader: Pointer; SampleStartTime: LargeInteger; SampleStopTime: LargeInteger) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnRawSample)(System::TObject \*Sender, Windows::PByte pSampleBuffer, **int** SampleBufferSize, **int** SampleDataLength, TFormatType FormatType, **void** \*pFormat,

**void** \*pBitmapInfoHeader, \_\_int64 SampleStartTime, \_\_int64 SampleStopTime);

### Description

Type of the [OnRawVideoSample](#) and [OnRawAudioSample](#) events.

**SampleBuffer:** pByte

Pointer to the buffer containing the sample data.

**SampleBufferSize:** Integer

Size of the buffer containing the sample data.

**SampleDataLength:** Integer

Length of the valid data in the buffer.

**FormatType:** type of the format, see [TFormatType](#)

**pFormat:** pointer

Pointer to the format whose type depends of the FormatType:

ft\_VideoInfo: pFormat is a pVideoInfoHeader pointer

ft\_VideoInfo2: pFormat is a pVideoInfoHeader2 pointer

ft\_MpegVideo: pFormat is a pMpegVideoInfo pointer

ft\_Mpeg2Video: pFormat is a pMpeg2VideoInfo pointer

ft\_WaveFormatEx: pFormat is a pWaveFormatEx pointer.

**SampleStartTime:** int64 (or Double for the OCX version)

Start time of the sample, expressed in 100 nano-seconds units.

**SampleEndTime:** int64 (or Double for the OCX version)

End time of the sample, expressed in 100 nano-seconds units.

### See Also

[OnRawAudioSample](#) [OnRawVideoSample](#) [RawAudioSampleCapture](#) [RawCaptureAsyncEvent](#)  
[RawVideoSampleCapture](#)

---

Created with the Standard Edition of HelpNDoc: [Free help authoring tool](#)

---

## TOnRecordingCompleted

### TOnRecordingCompleted

[Prev](#)

[Next](#)

The type of the [OnRecordingCompleted](#) event.

### Unit

[VidGrab](#)

### Declaration

TOnRecordingCompleted=**procedure**(Sender: TObject; FileName: **string**; Success: Boolean) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnRecordingCompleted)(System::TObject \*Sender, wchar\_t \*FileName, **bool** Success);

### Description

The type of the [OnRecordingCompleted](#) event.

**FileName:** full qualified path of the AVI file created.

**Success:** true upon success, false upon failure (e.g. insufficient disk space when reencoding).

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#)  
[TOnRecordingReadyToStart](#) [TRecordingMethod](#) [TSyncPreview](#) [AudioRecording](#) [AudioSyncAdjustment](#)

[AutoFileNameMinDigits](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[Encoder](#)
[SetInt](#)
[HoldRecording](#)
[IsRecordingPaused](#)
[Last\\_Recording\\_FileName](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Qt Help documentation made easy

## TOnRecordingReadyToStart

### TOnRecordingReadyToStart

[Prev](#)

[Next](#)

The type of the [OnRecordingReadyToStart](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnRecordingReadyToStart=**procedure**(Sender: TObject) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnRecordingReadyToStart)(System::TObject \*Sender);

#### Description

The type of the [OnRecordingReadyToStart](#) event.

**OkToStart:** assign true to OkToStart to allow the recording to start, false to cancel the recording.

#### See Also

[Recording methods and properties](#)
[TAVIMuxConfig](#)
[TASFDinterlaceMode](#)
[TAutoFileName](#)
[TOnRecordingCompleted](#)
[TRecordingMethod](#)
[TSyncPreview](#)
[AudioRecording](#)
[AudioSyncAdjustment](#)
[AutoFileNameMinDigits](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[Encoder](#)
[SetInt](#)
[HoldRecording](#)
[IsRecordingPaused](#)
[Last\\_Recording\\_FileName](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc

## TOnResizeVideo

### TOnResizeVideo

[Prev](#)

[Next](#)

The type of the [OnResizeVideo](#) event.

## Unit

[VidGrab](#)

## Declaration

TOnResizeVideo=**procedure**(Sender: TObject; SourceWidth, SourceHeight: LongInt) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnResizeVideo)(System::TObject \*Sender, **int** SourceWidth, **int** SourceHeight);

## Description

The type of the [OnResizeVideo](#) event.

**SourceWidth**: width of the video source

**SourceHeight**: height of the video source

## See Also

[TOnFrameProgress](#) [AutoRefreshPreview](#) [BackgroundColor](#) [Display](#) [AutoSize](#) [Display](#) [FullScreen](#) [Display](#) [SetLocation](#) [Display](#) [VideoPortEnabled](#) [ImageRatio](#) [InFrameProgressEvent](#) [IsVideoInterlaced](#) [IsVideoPortAvailable](#) [OnFrameProgress](#) [OnKeyPress](#) [OnPreviewStarted](#) [OnResizeVideo](#) [PausePreview](#) [PreviewZoomSize](#) [ResumePreview](#) [StartPreview](#) [StartSynchronized](#) [StopPreview](#) [Synchronized](#) [UseNearestVideoSize](#) [VideoHeight](#) [VideoRenderer](#) [VideoRendererPriority](#) [VideoWidth](#) [Visible](#)

Created with the Standard Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

## TOnSourceFileToDestFileCompleted

### TOnSourceFileToDestFileCompleted

[Prev](#)

[Next](#)

Type of the [OnCopyPreallocDataCompleted](#) and [OnReencodingCompleted](#) events.

## Unit

[VidGrab](#)

## Declaration

TOnSourceFileToDestFileCompleted=**procedure**(Sender: TObject; SourceFile, DestFile: **String**; Success: Boolean) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnSourceFileToDestFileCompleted)(System::TObject \*Sender, wchar\_t \*SourceFile, wchar\_t \*DestFile, **bool** Success);

## Description

Type of the [OnCopyPreallocDataCompleted](#) and [OnReencodingCompleted](#) events.

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

## TOnSourceFileToDestFileStarted

### TOnSourceFileToDestFileStarted

[Prev](#)

[Next](#)

Type of the [OnCopyPreallocDataStarted](#) and [OnReencodingStarted](#) events.

## Unit

[VidGrab](#)

## Declaration

TOnSourceFileToDestFileStarted=**procedure**(Sender: TObject; SourceFile, DestFile: **String**)**of object**;

```
typedef void __fastcall(__closure *TOnSourceFileToDestFileStarted)(System::TObject *Sender, wchar_t *SourceFile, wchar_t *DestFile);
```

#### Description

Type of the [OnCopyPreallocDataStarted](#) and [OnReencodingStarted](#) events.

---

Created with the Standard Edition of HelpNDoc: [Leave the tedious WinHelp HLP to CHM conversion process behind with HelpNDoc](#)

---

## TOnTextOverlayScrollingCompleted

### TOnTextOverlayScrollingCompleted

[Prev](#)
[Next](#)

Type of the [OnTextOverlayScrollingCompleted](#) event

#### Declaration

TOnTextOverlayScrollingCompleted=**procedure** (Sender: TObject; TextOverlayIndex: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnTextOverlayScrollingCompleted)(System::TObject *Sender, int TextOverlayIndex)
```

#### Description

Type of the [OnTextOverlayScrollingCompleted](#) event

---

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

---

## TOnThirdPartyFilterConnected

### TOnThirdPartyFilterConnected

[Prev](#)
[Next](#)

The type of the [OnThirdPartyFilterConnected](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnThirdPartyFilterConnected=**procedure**(Sender: TObject; Location: TThirdPartyFilterList; **Index**: LongInt; Intf: IUnknown) **of object**;

```
typedef void __fastcall(__closure *TOnThirdPartyFilterConnected)(System::TObject *Sender, TThirdPartyFilterList Location, int Index, _di_IUnknown Intf);
```

#### Description

The type of the [OnThirdPartyFilterConnected](#) event.

Returns the IUnknown interface of the connected third-party filter.

**Location**: list of the insertion point of the filter

**Index**: index of the filter in the Location list

**Intf**: returns the IUnknown interface of the third-party filter immediately after it has been connected, allowing to set/retrieve its properties programmatically.

#### See Also

[TThirdPartyFilterList](#) [OnThirdPartyFilterConnected](#) [ThirdPartyFilter](#) [AddToList](#) [ThirdPartyFilter](#) [ClearList](#) [ThirdPartyFilter](#) [Enable](#) [ThirdPartyFilter](#) [RemoveFromList](#) [ThirdPartyFilter](#) [ShowDialog](#)

---

Created with the Standard Edition of HelpNDoc: [Transform Your Word Doc into a Professional-Quality eBook with HelpNDoc](#)

---



## TOnThreadSync

### TOnThreadSync

[Prev](#)
[Next](#)

Type of the [OnThreadSync](#) event.

#### Description

Type of the [OnThreadSync](#) event.

Parameter:

ThreadSyncPoint: [TThreadSyncPoint](#)

#### See Also

[Opening a clip or an IP URL from a background thread without blocking the main thread](#) [Player features](#) [TThreadSyncPoint](#) [EnableThreadMode](#) [OnThreadSync](#)

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

## TOnTVChannelScanStarted

### TOnTVChannelScanStarted

[Prev](#)
[Next](#)

Type of the [OnTVChannelScanStarted](#) event.

#### Unit

[VidGrab](#)

#### Declaration

TOnTVChannelScanStarted=**procedure**(Sender: TObject; MinChannel: LongInt; MaxChannel: LongInt) **of object**;

```
typedef void __fastcall(__closure *TOnTVChannelScanStarted)(System::TObject *Sender, int MinChannel, int MaxChannel);
```

#### Description

Type of the [OnTVChannelScanStarted](#) event.

Reports the minimal and maximal channel that will be scanned for the current [country code](#) and [tuner input type](#).

#### See Also

[TTVChannelInfo](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

Created with the Standard Edition of HelpNDoc: [Maximize Your Productivity with a Help Authoring Tool](#)

## TOnTVChannelSelected

### TOnTVChannelSelected

[Prev](#)
[Next](#)

Type of the [OnTVChannelSelected](#) event.

**Unit**[VidGrab](#)**Declaration**

TOnTVChannelSelected=**procedure**(Sender: TObject; Channel: LongInt; Locked: Boolean; DefaultVideoFrequency: LongInt; OverriddenVideoFrequency: LongInt; TunerVideoFrequency: LongInt; TunerAudioFrequency: LongInt) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnTVChannelSelected)(System::TObject \*Sender, **int** Channel, **bool** Locked, **int** DefaultVideoFrequency, **int** OverriddenVideoFrequency, **int** TunerVideoFrequency, **int** TunerAudioFrequency);

**Description**

Type of the [OnTVChannelSelected](#) event.

**Channel:**

Reports the current TV channel in use

**Locked:**

Returns true if the horizontal sync is locked for this channel.

**DefaultVideoFrequency:**

Reports the default video frequency for the current TV country code / tuner input type.  
Can report -1 if a frequency override has been set and the default TV frequency is not saved in the registry.

**OverriddenVideoFrequency:**

Reports the overridden frequency for this channel, if it has been previously set by invoking TVSetChannelFrequencyOverrideVidGrab.TVideoGrabber.TVSetChannelFrequencyOverride.

**TunerVideoFrequency:**

Reports the current audio frequency, as reported by the TV tuner when selecting the TV channel.

**TunerAudioFrequency:**

Reports the current video frequency, as reported by the TV tuner when selecting the TV channel.

**See Also**

[TTVChannelInfo](#) [TOnTVChannelScanStarted](#) [TTunerInputType](#) [GetTVChannelInfo](#) [IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#) [OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#) [TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#) [TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#) [TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

---

Created with the Standard Edition of HelpNDoc: [Generate EPub eBooks with ease](#)

---

**TOnVideoCompressionSettings****TOnVideoCompressionSettings**[Prev](#)[Next](#)

Type of the [OnVideoCompressionSettings](#) event.

**Unit**[VidGrab](#)**Declaration**

TOnVideoCompressionSettings=**procedure**(Sender: TObject; CanKeyFrameRate: Boolean; CanPFFramesPerKeyFrame: Boolean; CanQuality: Boolean; CanWindowSize: Boolean; CanSpecifyDataRate: Boolean; DefaultFrameRate: LongInt; DefaultPFFramesPerKey: LongInt;

DefaultQuality: Double; DefaultWindowSize: LongWord; DefaultDataRate: LongWord) **of object**;

```
typedef void __fastcall(__closure *TOnVideoCompressionSettings)(System::TObject *Sender, bool CanKeyFrameRate, bool CanPFramesPerKeyFrame, bool CanQuality, bool CanWindowSize, bool CanSpecifyDataRate, int DefaultFrameRate, int DefaultPFramesPerKey, double DefaultQuality, unsigned DefaultWindowSize, unsigned DefaultDataRate);
```

### Description

Type of the [OnVideoCompressionSettings](#) event.

See "General video compression properties" in the [Software compression using codecs](#) chapter for more information about this event.

### See Also

[Recording methods and properties](#) [TCompressionType](#) [AudioCompressor](#) [AudioCompressorIndex](#) [AudioCompressorName](#) [AudioCompressors](#) [AudioCompressorsCount](#) [CompressionMode](#) [CompressionType](#) [GetVideoCompressionSettings](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [OnVideoCompressionSettings](#) [RefreshDevicesAndCompressorsLists](#) [SaveCompressorSettingsToDataString](#) [SetVideoCompressionDefaults](#) [SetVideoCompressionSettings](#) [VideoCompression](#) [KeyFrameRate](#) [VideoCompression\\_PFramesPerKeyFrame](#) [VideoCompression\\_Quality](#) [VideoCompression\\_WindowSize](#) [VideoCompressor](#) [VideoCompressorIndex](#) [VideoCompressorName](#) [VideoCompressors](#) [VideoCompressorsCount](#)

---

Created with the Standard Edition of HelpNDoc: [Effortlessly upgrade your WinHelp HLP help files to CHM with HelpNDoc](#)

---

## TOnVideoFromBitmapsNextFrameNeeded

### TOnVideoFromBitmapsNextFrameNeeded

[Prev](#)
[Next](#)

Type of the [OnVideoFromBitmapsNextFrameNeeded](#) event.

### Unit

[VidGrab](#)

### Declaration

TOnVideoFromBitmapsNextFrameNeeded=**procedure**(Sender: TObject; FirstSample: Boolean; **var** BitmapHandle: HBITMAP; **var** CanFreeBitmapHandle: Boolean; **var** BMPorJPEGFile: **String**; **var** EndOfData: Boolean) **of object**;

```
typedef void __fastcall(__closure *TOnVideoFromBitmapsNextFrameNeeded)(System::TObject *Sender, bool FirstSample, HBITMAP &BitmapHandle, bool &CanFreeBitmapHandle, wchar_t *&BMPorJPEGFile, bool &EndOfData);
```

### Description

Type of the [OnVideoFromBitmapsNextFrameNeeded](#) event.

### See Also

---

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

---

## TOnVideoKeyPress

### TOnVideoKeyPress

[Prev](#)
[Next](#)

Type of the [TOnKeyPress](#) event.

## Unit

[VidGrab](#)

### Declaration

TOnVideoKeyPress=**procedure**(Sender: TObject; VideoWindow: LongInt; **var** Key: Char; PhysicalKey: LongInt; ShiftState: TShiftState) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnVideoKeyPress)(System::TObject \*Sender, **int** VideoWindow, **char** &Key, **int** PhysicalKey, Classes::TShiftState ShiftState);

### Description

Type of the [TOnKeyPress](#) event.

### Note:

VideoWindow returns the number of the video window where the event occurred:

- 0 (by default) for single display,
- 0 or 1 when using dual display, according to the video window on which the event occurred.

---

Created with the Standard Edition of HelpNDoc: [Effortlessly Publish Your Word Document as an eBook](#)

---

## TOnVideoMouseMove

### TOnVideoMouseMove

[Prev](#)

[Next](#)

Type of the [OnMouseMove](#) event.

## Unit

[VidGrab](#)

### Declaration

TOnVideoMouseMove=**procedure**(Sender: TObject; VideoWindow: LongInt; Shift: TShiftState; X, Y: LongInt) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnVideoMouseMove)(System::TObject \*Sender, **int** VideoWindow, Classes::TShiftState Shift, **int** X, **int** Y);

### Description

Type of the [OnMouseMove](#) event.

### Note:

VideoWindow returns the number of the video window where the event occurred:

- 0 (by default) for single display,
- 0 or 1 when using dual display, according to the video window on which the event occurred.

---

Created with the Standard Edition of HelpNDoc: [Don't Let Unauthorized Users View Your PDFs: Learn How to Set Passwords](#)

---

## TOnVideoMouseUpDown

### TOnVideoMouseUpDown

[Prev](#)

[Next](#)

Type of the [OnMouseDown](#) and [OnMouseUp](#) events.

## Unit

[VidGrab](#)

### Declaration

TOnVideoMouseUpDown=**procedure**(Sender: TObject; VideoWindow: LongInt; Button: TMouseButton; Shift: TShiftState; X, Y: LongInt) **of object**;

**typedef void** \_\_fastcall(\_\_closure \*TOnVideoMouseUpDown)(System::TObject \*Sender, **int** VideoWindow, Controls::TMouseButton Button, Classes::TShiftState Shift, **int** X, **int** Y);

### Description

Type of the [OnMouseDown](#) and [OnMouseUp](#) events.

### Note:

VideoWindow returns the number of the video window where the event occurred:

- 0 (by default) for single display,
- 0 or 1 when using dual display, according to the video window on which the event occurred.

---

Created with the Standard Edition of HelpNDoc: [Effortlessly bring your documentation online with HelpNDoc](#)

---

## TONVIFDeviceInfo

### TONVIFDeviceInfo

[Prev](#)
[Next](#)

Type of parameter returned by the [ONVIFDeviceInfo](#) function

### Declaration

TONVIFDeviceInfo=(onv\_Manufacturer, onv\_Model, onv\_HardwareId, onv\_SerialNumber, onv\_FirmwareVersion, onv\_PTZInfo, onv\_PTZPresets);

typedef enum TONVIFDeviceInfo

onv\_Manufacturer=0,

onv\_Model=1,

onv\_HardwareId=2,

onv\_SerialNumber=3,

onv\_FirmwareVersion=4,

onv\_PTZInfo=5,

onv\_PTZPresets=6

TONVIFDeviceInfo;

### Description

Type of parameter returned by the [ONVIFDeviceInfo](#) function

TONVIFDeviceInfo = (onv\_Manufacturer, onv\_Model, onv\_HardwareId, onv\_SerialNumber, onv\_FirmwareVersion, onv\_PTZInfo, onv\_PTZPresets, onv\_MacAddress, onv\_AuxiliaryCommands, onv\_XMLReplay, onv\_XMLInfo);

### See Also

[IR Cut Filter of Axis cameras](#) [TDiscoveryCallbackStatus](#) [TOnONVIFDiscoveryCompletedNotification](#) [ExtraDLLPath](#) [GetLastErrorMessage](#) [OnONVIFDiscoveryCompletedNotification](#) [ONVIF\\_GetStr](#) [ONVIF\\_SetStr](#) [ONVIFCancelDiscovery](#) [ONVIFDeviceInfo](#) [ONVIFDiscoverCameras](#) [IPRange](#) [ONVIFDiscoverCameras\\_Multicast](#) [ONVIFEnumCamerasDiscovered](#) [ONVIFPTZGetLimits](#) [ONVIFPTZGetPosition](#) [ONVIFPTZPreset](#) [ONVIFPTZSendAuxiliaryCommand](#) [ONVIFPTZSetPosition](#) [ONVIFPTZStartMove](#) [ONVIFPTZStopMove](#) [ONVIFSnapShot](#) [OpenURLAsyncStatus](#)

---

Created with the Standard Edition of HelpNDoc: [Achieve Professional Documentation Results with a Help Authoring Tool](#)

---

## TOpenURLAsyncStatus

### TOpenURLAsyncStatus

[Prev](#)
[Next](#)

Type of the [OpenURLAsyncStatus](#) function

#### Description

TOpenURLAsyncStatus = (oas\_InProgress\_Connecting, oas\_InProgress\_Connected, oas\_Completed\_Success, oas\_Undefined, oas\_Completed\_Error);

**oas\_Undefined:** no asynchronous URL opening initiated yet

**oas\_InProgress\_Connecting:** asynchronous URL opening just started

**oas\_InProgress\_Connected:** URL connection successful, analyzis in progress

**oas\_Completed\_Success:** URL sucessfully connected

**oas\_Completed\_Error:** URL connection failed

Created with the Standard Edition of HelpNDoc: [5 Reasons Why a Help Authoring Tool is Better than Microsoft Word for Documentation](#)

## TPlayerState

### TPlayerState

[Prev](#)
[Next](#)

Type used by the [PlayerState](#) property and [OnPlayerStateChanged](#) events.

#### Declaration

TPlayerState=(ps\_Closed, ps\_Stopped, ps\_Paused, ps\_Playing, ps\_PlayingBackward, ps\_FastForwarding, ps\_FastRewinding);

**enum** TPlayerState ps\_Closed, ps\_Stopped, ps\_Paused, ps\_Playing, ps\_PlayingBackward, ps\_FastForwarding, ps\_FastRewinding ;

#### Description

Type used by the [PlayerState](#) property and [OnPlayerStateChanged](#) events.

The possible values are:

ps\_Closed  
ps\_Stopped  
ps\_Paused  
ps\_Playing  
ps\_PlayingBackward  
ps\_FastForwarding  
ps\_FastRewinding  
ps\_Downloading  
ps\_DownloadCompleted  
ps\_DownloadCancelled

Each value is greater than the previous one, so it is possible to test e.g. if the clip is playing in any direction by testing PlayerState >= ps\_Playing.

#### See Also

[Player features](#) [TOnPlayerStateChanged](#) [TOnPlayerBufferingData](#) [AudioChannelRenderMode](#) [AudioStreamNumber](#) [AutoStartPlayer](#) [AVIDuration](#) [AVIHeaderInfo](#) [AVIInfo](#) [AVIInfo2](#) [ClosePlayer](#) [FastForwardPlayer](#) [IsPlayerAudioStreamAvailable](#) [IsPlayerVideoStreamAvailable](#) [Last\\_Clip\\_Played](#) [MP4NeedsReindexing](#) [OnPlayerBufferingData](#) [OnPlayerEndOfStream](#) [OnPlayerStateChanged](#) [OnPlayerUpdateTrackbarPosition](#) [OpenPlayer](#) [OpenPlayerAtFramePositions](#) [OpenPlayerAtTimePositions](#) [PausePlayer](#) [PlayerAudioCodec](#) [PlayerAudioRendering](#) [PlayerDuration](#) [PlayerDVSize](#)

[PlayerFastSeekSpeedRatio](#) [PlayerFileName](#) [PlayerForcedCodec](#) [PlayerFrameCount](#) [PlayerFramePosition](#) [PlayerFrameRate](#) [PlayerFrameStep](#) [PlayerRefreshPausedDisplay](#) [PlayerRefreshPausedDisplayFrameRate](#) [PlayerSpeedRatio](#) [PlayerTimePosition](#) [PlayerTrackBar](#) [PlayerTrackBarScale](#) [PlayerTrackBarSynchron](#) [PlayerVideoCodec](#) [RewindPlayer](#) [RunPlayer](#) [RunPlayerBackwards](#) [ShowDialog](#) [SourceStream](#) [StopPlayer](#) [SynchronizationRole](#) [Synchronized](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoPlayableWhileRecording](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Full-featured Help generator](#)

## TPlaylist

### TPlaylist

[Prev](#)

[Next](#)

Type used by the [Playlist](#) function.

#### Declaration

TPlaylist=(pl\_Add, pl\_Remove, pl\_Clear, pl\_Loop, pl\_NoLoop, pl\_Play, pl\_Stop, pl\_Next, pl\_Previous, pl\_SortAlpha, pl\_SortRevAlpha, pl\_Random, pl\_Sequential);

enum TPlaylist pl\_Add, pl\_Remove, pl\_Clear, pl\_Loop, pl\_NoLoop, pl\_Play, pl\_Stop, pl\_Next, pl\_Previous, pl\_SortAlpha, pl\_SortRevAlpha, pl\_Random, pl\_Sequential ;

#### Description

See the [Playlist](#) function.

#### See Also

[Video formats](#) [GetPlaylist](#) [IsPlaylistActive](#) [OnPlayerEndOfPlaylist](#) [Playlist](#) [PlaylistIndex](#)

Created with the Standard Edition of HelpNDoc: [Revolutionize Your CHM Help File Output with HelpNDoc](#)

## TPointGreyConfig

### TPointGreyConfig

[Prev](#)

[Next](#)

Type used by the [PointGreyConfig](#) function.

#### Declaration

TPointGreyConfig=(pgr\_SetRegister, pgr\_GetRegister, pgr\_SetBufferSize, pgr\_GetBufferSize, pgr\_SetFormat, pgr\_GetFormat)

enum TPointGreyConfig pgr\_SetRegister, pgr\_GetRegister, pgr\_SetBufferSize, pgr\_GetBufferSize, pgr\_SetFormat, pgr\_GetFormat ;

#### Description

Possible values:

pgr\_SetRegister  
pgr\_GetRegister  
pgr\_SetBufferSize  
pgr\_GetBufferSize  
pgr\_SetFormat  
pgr\_GetFormat

See the [PointGreyConfig](#) function.

Created with the Standard Edition of HelpNDoc: [Free Kindle producer](#)



## TRawSampleCaptureLocation

### TRawSampleCaptureLocation

[Prev](#)

[Next](#)

Type of the [RawSampleCaptureLocation](#) property.

#### Declaration

TRawSampleCaptureLocation=(rl\_SourceFormat, rl\_AfterCompression);

**enum** TRawSampleCaptureLocationrl\_SourceFormat, rl\_AfterCompression ;

#### Description

#### Value Meaning

nm\_rl\_SourceFormat The sample capture is inserted on the capture device output.

nm\_rl\_AfterCompression The sample capture is inserted after the audio or video compressor, when recording and compressing on the fly.

Created with the Standard Edition of HelpNDoc: [Produce electronic books easily](#)

## TRecordingMethod

### TRecordingMethod

[Prev](#)

[Next](#)

Type of the [RecordingMethod](#) property.

#### Unit

[VidGrab](#)

#### Declaration

TRecordingMethod=(rm\_AVI, rm\_ASF, rm\_SendToDV, rm\_MKV, rm\_FLV, rm\_MP4, rm\_WebM, rm\_MPG, rm\_Multiplexer, rm\_MOV, rm\_TS);

**enum** TRecordingMethod rm\_AVI, rm\_ASF, rm\_SendToDV, rm\_MKV, rm\_FLV, rm\_MP4, rm\_WebM, rm\_MPG, rm\_Multiplexer, rm\_MOV, rm\_TS ;

#### Description

#### Value Meaning

rm\_AVI the recording will create an AVI or MPEG file,

rm\_ASF the recording will create an ASF file,

rm\_SendToDV the audio/video streams will be sent to the DV device.

rm\_MKV similar to rm\_AVI, but the recording will create a MKV file, see [AVI and MKV recording](#) (the Matroska muxer must be installed, it can be downloaded at <http://www.matroska.org/>)

rm\_FLV the recording will create a Flash video in FLV format, see [FLV recording](#)

rm\_MP4 the recording will create a .MP4 video clip, see [MP4 recording](#)

rm\_WebM the recording will create a .WebM video clip, see [WebM recording](#)

rm\_MPG the recording will create a .mpg file (MPEG)

rm\_Multiplexer used to record by using another third-party multiplexer specified with

[SetMultiplexerFilterByName](#)

rm\_MOV the recording will create a .MOV file, use the same settings than for [MP4 recording](#)

rm\_TS the recording will create a .TS file (Transport Stream). Useful if the capture device outputs a transport stream and [RecordingInNativeFormat=true](#)

rm\_H264 the recording will create a .H264 video file (no audio)

rm\_MP3 the recording will create a .MP3 audio clip

rm\_WMA the recording will create a .WMA audio clip

rm\_WAV the recording will create a .WAV audio clip

#### See Also

[Recording methods and properties](#)
[TAVIMuxConfig](#)
[TASFDeinterlaceMode](#)
[TAutoFileName](#)
[TOnRecordingCompleted](#)
[TOnRecordingReadyToStart](#)
[TSyncPreview](#)
[AudioRecording](#)
[AudioSyncAdjustment](#)
[AutoFileNameMinDigits](#)
[AVIDurationUpdated](#)
[AVIFormatOpenDML](#)
[AVIHeaderInfo](#)
[AVIInfo](#)
[Encoder](#)
[SetInt](#)
[HoldRecording](#)
[IsRecordingPaused](#)
[Last\\_Recording\\_FileName](#)
[OnBacktimedFramesCountReached](#)
[OnCopyPreallocDataCompleted](#)
[OnCopyPreallocDataProgress](#)
[OnCopyPreallocDataStarted](#)
[OnCreatePreallocFileCompleted](#)
[OnCreatePreallocFileStarted](#)
[OnDiskFull](#)
[OnRecordingCompleted](#)
[OnRecordingPaused](#)
[OnRecordingReadyToStart](#)
[OnRecordingStarted](#)
[OnReencodingCompleted](#)
[OnReencodingStarted](#)
[PauseRecording](#)
[PreallocCapFileCopiedAfterRecording](#)
[PreallocCapFileEnabled](#)
[PreallocCapFileName](#)
[PreallocCapFileSizeInMB](#)
[RecordingBacktimedFramesCount](#)
[RecordingCanPause](#)
[RecordingDuration](#)
[RecordingFileName](#)
[RecordingFileSizeMaxInMB](#)
[RecordingFourCC](#)
[RecordingHeight](#)
[RecordingInNativeFormat](#)
[RecordingKBytesWrittenToDisk](#)
[RecordingMethod](#)
[RecordingSize](#)
[RecordingTimer](#)
[RecordingTimerInterval](#)
[RecordingWidth](#)
[ResumeRecording](#)
[SaveCompressorSettingsToDataString](#)
[SetMultiplexerFilterByName](#)
[StartAudioRecording](#)
[StartRecording](#)
[StartSynchronized](#)
[StopRecording](#)
[StoragePath](#)
[Synchronized](#)
[SyncPreview](#)
[VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Save time and frustration with HelpNDoc's WinHelp HLP to CHM conversion feature](#)

## TRecordingTimer

### TRecordingTimer

[Prev](#)

[Next](#)

Type of the [RecordingTimer](#) property.

#### Declaration

TRecordingTimer=(rt\_Disabled, rt\_RecordToNewFile, rt\_StopRecording, rt\_StartRecording);

**enum** TRecordingTimer rt\_Disabled, rt\_RecordToNewFile, rt\_StopRecording, rt\_StartRecording ;

#### Description

Type of the [RecordingTimer](#) property.

Possible values:

rt\_Disabled  
rt\_RecordToNewFile  
rt\_StopRecording  
rt\_StartRecording  
rt\_FrameCapture

Created with the Standard Edition of HelpNDoc: [Effortlessly Create Professional Documentation with HelpNDoc's Clean UI](#)

## TRegistryRoot

### TRegistryRoot

[Prev](#)

[Next](#)

Type of the Rootkey global property.

#### Unit

[VidGrab](#)

#### Declaration

TRegistryRoot=(RR\_HKEY\_CURRENT\_USER, RR\_HKEY\_LOCAL\_MACHINE);

**enum** TRegistryRoot RR\_HKEY\_CURRENT\_USER, RR\_HKEY\_LOCAL\_MACHINE ;

## Description

### Value Meaning

RR\_HKEY\_CURRENT\_USER  
RR\_HKEY\_LOCAL\_MACHINE

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

## TRGBSelector

### TRGBSelector

[Prev](#)

[Next](#)

Used to select the R, G or B color.

### Declaration

TRGBSelector=(rs\_Red, rs\_Green, rs\_Blue);

**enum** RGBSelector rs\_Red, rs\_Green, rs\_Blue;

### Description

Type of the RGBSelector parameter in the [MotionDetector\\_CellColorIntensity](#) and [MotionDetector\\_GlobalColorIntensity](#) functions.  
Used to select one of the R, G or B colors.

Created with the Standard Edition of HelpNDoc: [Free EPub producer](#)

## TStreamType

### TStreamType

[Prev](#)

[Next](#)

TStreamType

### Description

TStreamType = (st\_Video, st\_Audio);

Created with the Standard Edition of HelpNDoc: [Free EPub and documentation generator](#)

## TSynchronizationRole

### TSynchronizationRole

[Prev](#)

[Next](#)

Synchronization role

### Declaration

TSynchronizationRole =(sr\_Master, sr\_Slave);

**enum** TSynchronizationRole sr\_Master, sr\_Slave ;

### Description

Type of the [SynchronizationRole](#) function

Created with the Standard Edition of HelpNDoc: [Full-featured EBook editor](#)

## TSyncPreview

### TSyncPreview

[Prev](#)

[Next](#)

Type of the [SyncPreview](#) property.

### Unit

[VidGrab](#)

### Declaration

TSyncPreview=(sp\_Auto, sp\_Disabled, sp\_Enabled);

**enum** TSyncPreview sp\_Auto, sp\_Disabled, sp\_Enabled ;

### Description

#### Value Meaning

sp\_Auto Automatically selected

sp\_Disabled the sync of the audio rendering and video preview is disabled (this can cause time shift between the audio rendering and the video displayed

sp\_Enabled the sync of the audio rendering and video display is enabled (this can cause dropped frames and/or audio/video sync problems in the recorded AVI)

### See Also

[Recording methods and properties](#) [TAVIMuxConfig](#) [TASFDeinterlaceMode](#) [TAutoFileName](#) [TOnRecordingCompleted](#) [TOnRecordingReadyToStart](#) [TRecordingMethod](#) [AudioRecording](#) [AudioSyncAdjustment](#) [AutoFileNameMinDigits](#) [AVIDurationUpdated](#) [AVIFormatOpenDML](#) [AVIHeaderInfo](#) [AVIInfo](#) [Encoder](#) [SetInt](#) [HoldRecording](#) [IsRecordingPaused](#) [Last\\_Recording\\_FileName](#) [OnBacktimedFramesCountReached](#) [OnCopyPreallocDataCompleted](#) [OnCopyPreallocDataProgress](#) [OnCopyPreallocDataStarted](#) [OnCreatePreallocFileCompleted](#) [OnCreatePreallocFileStarted](#) [OnDiskFull](#) [OnRecordingCompleted](#) [OnRecordingPaused](#) [OnRecordingReadyToStart](#) [OnRecordingStarted](#) [OnReencodingCompleted](#) [OnReencodingStarted](#) [PauseRecording](#) [PreallocCapFileCopiedAfterRecording](#) [PreallocCapFileEnabled](#) [PreallocCapFileName](#) [PreallocCapFileSizeInMB](#) [RecordingBacktimedFramesCount](#) [RecordingCanPause](#) [RecordingDuration](#) [RecordingFileName](#) [RecordingFileSizeMaxInMB](#) [RecordingFourCC](#) [RecordingHeight](#) [RecordingInNativeFormat](#) [RecordingKBytesWrittenToDisk](#) [RecordingMethod](#) [RecordingSize](#) [RecordingTimer](#) [RecordingTimerInterval](#) [RecordingWidth](#) [ResumeRecording](#) [SaveCompressorSettingsToDataString](#) [SetMultiplexerFilterByName](#) [StartAudioRecording](#) [StartRecording](#) [StartSynchronized](#) [StopRecording](#) [StoragePath](#) [Synchronized](#) [SyncPreview](#) [VideoPlayableWhileRecording](#)

Created with the Standard Edition of HelpNDoc: [Easily create EPub books](#)

## TTextOverlayGradientMode

### TTextOverlayGradientMode

[Prev](#)

[Next](#)

Used to enable the gradient mode of the text overlays

### Declaration

TTextOverlayGradientMode=(gm\_Disabled, gm\_Horizontal, gm\_Vertical, gm\_ForwardDiagonal, gm\_BackwardDiagonal);

**enum** TTextOverlayGradientMode gm\_Disabled, gm\_Horizontal, gm\_Vertical, gm\_ForwardDiagonal, gm\_BackwardDiagonal ;

### Description

Enables enable the gradient mode of the text overlays, and select the orientation

#### gm\_Disabled:

gradient mode disabled

#### gm\_Horizontal:

gradient applied from left to right

**gm\_Vertical:**

gradient applied for top to bottom

**gm\_ForwardDiagonal:**

gradient applied from top-left to right-bottom

**gm\_BackwardDiagonal:**

gradient applied from top-right to left-bottom

---

Created with the Standard Edition of HelpNDoc: [From Word to ePub or Kindle eBook: A Comprehensive Guide](#)

---

## TThirdPartyFilterList

### TThirdPartyFilterList

[Prev](#)

[Next](#)

Type used by the [third-party](#) functions.

**Declaration**

TThirdPartyFilterList=(tpf\_VideoSource, tpf\_VideoPreview, tpf\_VideoRecording, tpf\_AudioSource, tpf\_AudioRendering, tpf\_AudioRecording, tpf\_VideoRenderer, tpf\_AudioRenderer);

**enum** TThirdPartyFilterList tpf\_VideoSource, tpf\_VideoPreview, tpf\_VideoRecording, tpf\_AudioSource, tpf\_AudioRendering, tpf\_AudioRecording, tpf\_VideoRenderer, tpf\_AudioRenderer ;

**Description**

Type used by the [third-party](#) functions to specify the insertion point of the third-party filter:

### THIRD-PARTY FILTERS USED AS VIDEO AND/OR AUDIO SOURCE

**tpf\_ThirdPartyVideoSource:** the filter will be used as a video source if [VideoSource](#) = vs\_ThirdPartyFilter (and as an audio source if it exposes also an audio pin and [AudioSource](#) = as\_Default )

**tpf\_ThirdPartyAudioSource:** the filter will be used as an audio source if [VideoSource](#) = vs\_ThirdPartyFilter (if you are using also another third-party filter as video source and it exposes an audio pin, set [AudioSource](#) = as\_UseExternalAudio to ignore the audio pin of the video source filter and use the tpf\_ThirdPartyAudioSource filter as audio source)

### THIRD-PARTY FILTERS INSERTED IN THE NORMAL DIRECTSHOW GRAPH

**Note:** excepted the tpf\_VideoRenderer and the tpf\_AudioRenderer (that are must be renderers without output pins), all the filters must be "inline" filters with input pins and output pins.

**tpf\_VideoSource:** the filter is inserted before the video recording/preview switching, so the effect is applied to both video preview and video recording.

**tpf\_VideoPreview:** the filter is inserted on the video preview stream, after the recording/preview splitter. The effect is applied only to the video preview.

**tpf\_VideoRecording:** the filter is inserted on the video recording stream, after the recording/preview splitter. The effect is applied only to the video recording.

**tpf\_AudioSource:** the filter is inserted before the audio recording/rendering switching, so the effect is applied to both audio rendering and audio recording.

**tpf\_AudioRendering:** the filter is inserted on the audio rendering stream, after the recording/rendering splitter. The effect is applied only to audio rendering.

**tpf\_AudioRecording:** the filter is inserted on the audio recording stream, recording/rendering splitter. The effect is applied only to audio rendering.

**tpf\_VideoRenderer:** the filter IS A VIDEO RENDERER FILTER. It is inserted at the end of the preview stream.

**tpf\_AudioRenderer:** the filter IS AN AUDIO RENDERER FILTER. It is inserted at the end of the audio rendering stream.

**tpf\_ThirdPartyVideoSource:** the filter IS A VIDEO SOURCE

**tpf\_ThirdPartyAudioSource:** the filter IS AN AUDIO SOURCE

**tpf\_AddToGraph:** the filter is added to the graph in order to be used if compatible

#### See Also

[TOnThirdPartyFilterConnected](#) [OnThirdPartyFilterConnected](#) [ThirdPartyFilter\\_AddToList](#)  
[ThirdPartyFilter\\_ClearList](#) [ThirdPartyFilter\\_Enable](#) [ThirdPartyFilter\\_RemoveFromList](#)  
[ThirdPartyFilter\\_ShowDialog](#)

---

Created with the Standard Edition of HelpNDoc: [Elevate Your CHM Help Files with HelpNDoc's Advanced Customization Options](#)

---

## TThreadSyncPoint

### TThreadSyncPoint

[Prev](#)
[Next](#)

TThreadSyncPoint

#### Description

TThreadSyncPoint = (tsp\_Starting\_CanSetParent, tsp\_Started\_CanSetParent, tsp\_Stop\_UnSetParent);

**tsp\_Starting\_CanSetParent, tsp\_Started\_CanSetParent:**

the video window parent must be set from one of these events

**tsp\_Stop\_UnSetParent:**

the video window parent must be unset from this event

#### See Also

[Opening a clip or an IP URL from a background thread without blocking the main thread](#) [Player features](#)  
[TOnThreadSync](#) [EnableThreadMode](#) [OnThreadSync](#)

---

Created with the Standard Edition of HelpNDoc: [Easily create EBooks](#)

---

## TTrackbarAction

### TTrackbarAction

[Prev](#)
[Next](#)

TTrackbarAction type

#### Declaration

TTrackbarAction=(tba\_MouseDown, tba\_MouseUp, tba\_KeyDown, tba\_KeyUp);

**enum** TTrackbarAction tba\_MouseDown, tba\_MouseUp, tba\_KeyDown, tba\_KeyUp ;

#### Description

Type used by the [NotifyPlayerTrackbarAction](#) procedure.

See the player's [Trackbar](#) chapter for more information.

---

Created with the Standard Edition of HelpNDoc: [Maximize Your CHM Help File Capabilities with HelpNDoc](#)

---

## TTriState

### TTriState

[Prev](#)
[Next](#)

Like a boolean, but with an third "undefined" state.

#### Unit

[VidGrab](#)

#### Declaration

```
TTriState=(ts_Undefined, ts_False, ts_True);
```

```
enum TTriState ts_Undefined, ts_False, ts_True ;
```

#### Description

##### Value

ts\_Undefined

ts\_False

ts\_True

---

Created with the Standard Edition of HelpNDoc: [Revolutionize Your Documentation Output with a Help Authoring Tool](#)

---

## TTunerMode

### TTunerMode

[Prev](#)[Next](#)

Type of the TunerMode property

#### Description

Type of the TunerMode property.

The possible values are:

tm\_TVTuner

tm\_FMRadioTuner

tm\_AMRadioTuner

tm\_DigitalSatelliteTuner

---

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

---

## TTVChannelInfo

### TTVChannelInfo

[Prev](#)[Next](#)

Type passed to the [GetTVChannelInfo](#) function.

#### Declaration

```
TTVChannelInfo=(tci_Channel, tci_DefaultVideoFrequency, tci_OverriddenVideoFrequency,  
tci_TunerVideoFrequency, tci_TunerAudioFrequency, tci_Locked);
```

```
enum TTVChannelInfo tci_Channel, tci_DefaultVideoFrequency, tci_OverriddenVideoFrequency,  
tci_TunerVideoFrequency, tci_TunerAudioFrequency, tci_Locked ;
```

#### Description

Type of the desired channel info that will be returned by [GetTVChannelInfo](#).

##### tci\_Channel:

Reports the current TV channel in use (previewing or capturing to AVI).



**tci\_DefaultVideoFrequency:**

Reports the default video frequency for the current TV country code / tuner input type.  
Can report -1 if a frequency override has been set and the default TV frequency is not saved in the registry.

**tci\_OverriddenVideoFrequency:**

Reports the overridden frequency for this channel, if it has been previously set by invoking TVSetChannelFrequencyOverrideVidGrab.TVideoGrabber.TVSetChannelFrequencyOverride.

**tci\_TunerVideoFrequency:**

Reports the current audio frequency, as reported by the TV tuner when selecting the TV channel.

**tci\_TunerAudioFrequency, :**

Reports the current video frequency, as reported by the TV tuner when selecting the TV channel.

**tci\_Locked:**

Returns 1 if the horizontal sync is locked for this channel, otherwise returns 0.

**See Also**

[TOnTVChannelScanStarted](#) [TOnTVChannelSelected](#) [TTunerInputType](#) [GetTVChannelInfo](#)  
[IsHorizontalSyncLocked](#) [IsTVAudioAvailable](#) [IsTVAutoTuneRunning](#) [IsTVTunerAvailable](#)  
[OnTVChannelScanCompleted](#) [OnTVChannelScanStarted](#) [OnTVChannelSelected](#) [TunerFrequency](#)  
[TunerMode](#) [TVChannel](#) [TVClearFrequencyOverrides](#) [TVCountryCode](#) [TVGetMinMaxChannels](#)  
[TVSetChannelFrequencyOverride](#) [TVStartAutoScan](#) [TVStopAutoScan](#) [TVTunerInputType](#)  
[TVUseFrequencyOverrides](#) [VCRHorizontalLocking](#)

---

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Intuitive Interface](#)

---

## Tv360\_Angle

### Tv360\_Angle

[Prev](#)

[Next](#)

Tv360\_Angle

**Declaration**

Tv360\_Angle=(v360\_fov\_Diagonal, v360\_fov\_Horizontal, v360\_fov\_Vertical);

**enum** Tv360\_Angle v360\_fov\_Diagonal, v360\_fov\_Horizontal, v360\_fov\_Vertical ;

**Description**

Tv360\_Angle enumeration

---

Created with the Standard Edition of HelpNDoc: [Elevate Your Help Documentation with a Help Authoring Tool](#)

---

## Tv360\_InOut

### Tv360\_InOut

[Prev](#)

[Next](#)

Tv360\_InOut

**Declaration**

Tv360\_InOut=(v360\_in, v360\_out);

**enum** Tv360\_InOut v360\_in, v360\_out ;

#### Description

Tv360\_InOut enumeration

---

Created with the Standard Edition of HelpNDoc: [Make CHM Help File Creation a Breeze with HelpNDoc](#)

---

## Tv360\_Interpolation

### Tv360\_Interpolation

[Prev](#)[Next](#)

Tv360\_Interpolation

#### Declaration

Tv360\_Interpolation=(ipl\_Default, ipl\_Bilinear, ipl\_Nearest, ipl\_Lagrange9, ipl\_Bicubic, ipl\_Lanczos, ipl\_Spline16, ipl\_Gaussian, ipl\_Mitchell);

**enum** Tv360\_Interpolation ipl\_Default, ipl\_Bilinear, ipl\_Nearest, ipl\_Lagrange9, ipl\_Bicubic, ipl\_Lanczos, ipl\_Spline16, ipl\_Gaussian, ipl\_Mitchell ;

#### Description

Tv360\_Interpolation enumeration

---

Created with the Standard Edition of HelpNDoc: [Easily create Qt Help files](#)

---

## TV360\_MouseAction

### TV360\_MouseAction

[Prev](#)[Next](#)

TV360\_MouseAction

#### Declaration

TV360\_MouseAction=(ma\_Disabled, ma\_MouseUp, ma\_MouseMove);

**enum** TV360\_MouseAction ma\_Disabled, ma\_MouseUp, ma\_MouseMove ;

#### Description

TV360\_MouseAction enumeration

---

Created with the Standard Edition of HelpNDoc: [Easily share your documentation with the world through a beautiful website](#)

---

## Tv360\_Projection

### Tv360\_Projection

[Prev](#)[Next](#)

Tv360\_Projection

**Declaration**

```
Tv360_Projection=(
ipp_Equirectangular,
ipp_Cubemap_3_2,
ipp_Cubemap_6_1,
ipp_Equiangular,
ipp_Flat,
ipp_Dual_fisheye,
ipp_Barrel,
ipp_Cubemap_1_6,
ipp_Stereographic,
ipp_Mercator,
ipp_Ball,
ipp_Hammer,
ipp_Sinusoidal,
ipp_Fisheye,
ipp_Pannini,
ipp_Cylindrical,
ipp_Perspective,
ipp_Tetrahedron,
ipp_Barrel_split,
ipp_Tspyramid,
ipp_Hequirectangular,
ipp_Equisolid,
ipp_Orthographic,
ipp_Octahedron
);
```

**enum Tv360\_Projection**

```
ipp_Equirectangular,
ipp_Cubemap_3_2,
ipp_Cubemap_6_1,
ipp_Equiangular,
ipp_Flat,
ipp_Dual_fisheye,
ipp_Barrel,
ipp_Cubemap_1_6,
ipp_Stereographic,
ipp_Mercator,
ipp_Ball,
ipp_Hammer,
ipp_Sinusoidal,
ipp_Fisheye,
ipp_Pannini,
ipp_Cylindrical,
ipp_Perspective,
ipp_Tetrahedron,
ipp_Barrel_split,
ipp_Tspyramid,
ipp_Hequirectangular,
ipp_Equisolid,
ipp_Orthographic,
ipp_Octahedron
};
```

**Description**

Tv360\_Projection enumeration

## Tv360\_StereoFormat

### Tv360\_StereoFormat

[Prev](#)
[Next](#)

Tv360\_StereoFormat

#### Declaration

Tv360\_StereoFormat=(sf\_2DMono, sf\_SideBySide, sf\_TopBottom);

**enum** Tv360\_StereoFormat sf\_2DMono, sf\_SideBySide, sf\_TopBottom ;

#### Description

Tv360\_StereoFormat enumeration

Created with the Standard Edition of HelpNDoc: [Write EPub books for the iPad](#)

## TVideoAlignment

### TVideoAlignment

[Prev](#)
[Next](#)

Type of the [TextOverlay\\_VideoAlignment](#) and [ImageOverlay\\_VideoAlignment](#) properties

#### Description

TVideoAlignment = (oa\_LeftTop, oa\_CenterTop, oa\_RightTop, oa\_LeftCenter, oa\_Center, oa\_RightCenter, oa\_LeftBottom, oa\_CenterBottom, oa\_RightBottom);

Created with the Standard Edition of HelpNDoc: [Elevate Your Documentation Process with HelpNDoc's Advanced Features](#)

## TVideoDeinterlacing

### TVideoDeinterlacing

[Prev](#)
[Next](#)

Type of the [VideoProcessing\\_Deinterlacing](#) property.

#### Unit

[VidGrab](#)

#### Declaration

TVideoDeinterlacing=(di\_Disabled, di\_HalfSize, di\_FullSize);

**enum** TVideoDeinterlacing di\_Disabled, di\_HalfSize, di\_FullSize ;

#### Description

**di\_Disabled** : deinterlacing is disabled

**di\_HalfSize** : half-size deinterlacing

**di\_FullSize** : full size deinterlacing.

**di\_DScaler** : uses the DScaler deinterlacer

**di\_AVISynth** : uses the AVISynth deinterlacer (AVISynth must be installed)

**di\_FFDSHOW** : uses the FFDSHOW deinterlacer (FFDSHOW must be installed)

**di\_ThirdPartyDeinterlacer**: uses the deinterlacer specified to the ThirdPartyDeinterlacer property

See the [Deinterlacing](#) chapter for more information.

#### See Also

[MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrievalInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#) [TVideoRotation](#)

Created with the Standard Edition of HelpNDoc: [Easily create Web Help sites](#)

## TVideoRenderer

### TVideoRenderer

[Prev](#)
[Next](#)

Type of the [VideoRenderer](#) property.

#### Declaration

TVideoRenderer=(vr\_AutoSelect, vr\_VMR9, vr\_VMR7, vr\_StandardRenderer, vr\_OverlayRenderer, vr\_None);

**enum** TVideoRenderer vr\_AutoSelect, vr\_VMR9, vr\_VMR7, vr\_StandardRenderer, vr\_OverlayRenderer, vr\_None ;

#### Description

TVideoRenderer = (vr\_AutoSelect, vr\_EVR, vr\_VMR9, vr\_VMR7, vr\_StandardRenderer, vr\_OverlayRenderer, vr\_RecordingPriority, vr\_None, vr\_madVR, vr\_DatasteadRealtimeRenderer);  
vr\_AutoSelect: selected automatically among the values below, depending on the context

**vr\_None:** no video renderer

**vr\_RecordingPriority:** suitable for general use

**vr\_DatasteadRealtimeRenderer:** low latency, real-time renderer. Requires the [Datatastead RTSP/RTMP/HTTP/ONVIF Source filter](#) to be intalled.

**vr\_StandardRenderer:** standard DirectShow renderer

**vr\_VMR7:** VMR7 DirectShow renderer

**vr\_VMR9:** VMR9 DirectShow renderer, requires DirectX to be enabled

**vr\_EVR:** Enhanced DirectShow renderer, good quality but higher CPU consumption

**vr\_OverlayRenderer:** deprecated

**vr\_madVR:** deprecated

#### See Also

[AdjustPixelAspectRatio](#) [Display\\_Active](#) [Display\\_AlphaBlendEnabled](#) [Display\\_AlphaBlendValue](#) [Display\\_AutoSize](#) [Display\\_Embedded](#) [Display\\_FullScreen](#) [Display\\_Height](#) [Display\\_Left](#) [Display\\_Monitor](#) [Display\\_MouseMovesWindow](#) [Display\\_PanScanRatio](#) [Display\\_SetLocation](#) [Display\\_StayOnTop](#) [Display\\_Top](#) [Display\\_TransparentColorEnabled](#) [Display\\_TransparentColorValue](#) [Display\\_VideoHeight](#) [Display\\_VideoPortEnabled](#) [Display\\_VideoWidth](#) [Display\\_VideoWindowHandle](#) [Display\\_Width](#) [DualDisplay\\_Active](#) [DualDisplay\\_AlphaBlendEnabled](#) [DualDisplay\\_AlphaBlendValue](#) [DualDisplay\\_AutoSize](#) [DualDisplay\\_Embedded](#) [DualDisplay\\_FullScreen](#) [DualDisplay\\_Height](#) [DualDisplay\\_Left](#) [DualDisplay\\_Monitor](#) [DualDisplay\\_MouseMovesWindow](#) [DualDisplay\\_PanScanRatio](#) [DualDisplay\\_StayOnTop](#) [DualDisplay\\_Top](#) [DualDisplay\\_TransparentColorEnabled](#) [DualDisplay\\_TransparentColorValue](#) [DualDisplay\\_VideoHeight](#) [DualDisplay\\_VideoPortEnabled](#) [DualDisplay\\_VideoWidth](#) [DualDisplay\\_VideoWindowHandle](#) [DualDisplay\\_Visible](#) [DualDisplay\\_Width](#) [IsVideoPortAvailable](#) [Monitor\\_Primary](#) [Index\\_MonitorBounds](#) [MonitorsCount](#) [OnLeavingFullScreen](#) [SetParentWindow](#) [SetWindowTransparency](#) [VideoDoubleBuffered](#) [VideoHeight](#) [PreferredAspectRatio](#) [VideoRendererExternal](#) [VideoRendererExternalIndex](#) [VideoRendererPriority](#) [VideoVisibleWhenStopped](#) [VideoWidth](#) [PreferredAspectRatio](#)

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## TVideoRendererPriority

### TVideoRendererPriority

[Prev](#)
[Next](#)

Type of the [VideoRendererPriority](#) property

### Description

TVideoRendererPriority = (vrp\_Speed, vrp\_Quality, vrp\_Auto);

vrp\_Speed: favors performance and smoothness. The video quality may be lower especially when the video window is downsized.

vrp\_Quality: favors video quality. Requires more CPU.

vrp\_Auto: vrp\_Speed or vrp\_Quality is selected automatically depending on the context.

Created with the Standard Edition of HelpNDoc: [Modernize your help files with HelpNDoc's WinHelp HLP to CHM conversion tool](#)

## TVideoRotation

### TVideoRotation

[Prev](#)
[Next](#)

Type of the [VideoProcessing\\_Rotation](#) property.

### Unit

[VidGrab](#)

### Declaration

TVideoRotation=(rt\_0\_deg, rt\_90\_deg, rt\_180\_deg, rt\_270\_deg, rt\_0\_deg\_mirror, rt\_90\_deg\_mirror, rt\_180\_deg\_mirror, rt\_270\_deg\_mirror);

**enum** TVideoRotation rt\_0\_deg, rt\_90\_deg, rt\_180\_deg, rt\_270\_deg, rt\_0\_deg\_mirror, rt\_90\_deg\_mirror, rt\_180\_deg\_mirror, rt\_270\_deg\_mirror ;

### Description

Type of the [VideoProcessing\\_Rotation](#) property.

See [VideoProcessing\\_Rotation](#).

### See Also

[TVideoDeinterlacing](#) [MP4NeedsReindexing](#) [OnFrameBitmap](#) [RetrieveInitialXYAfterRotation](#) [VideoProcessing\\_Brightness](#) [VideoProcessing\\_Contrast](#) [VideoProcessing\\_Deinterlacing](#) [VideoProcessing\\_FlipHorizontal](#) [VideoProcessing\\_FlipVertical](#) [VideoProcessing\\_GrayScale](#) [VideoProcessing\\_Hue](#) [VideoProcessing\\_InvertColors](#) [VideoProcessing\\_Pixellization](#) [VideoProcessing\\_Rotation](#) [VideoProcessing\\_RotationCustomAngle](#) [VideoProcessing\\_Saturation](#)

Created with the Standard Edition of HelpNDoc: [Transform Your CHM Help File Creation Process with HelpNDoc](#)

## TVideoSource

### TVideoSource

[Prev](#)
[Next](#)

Type of the [VideoSource](#) property.

### Unit

[VidGrab](#)

### Declaration

TVideoSource=(vs\_VideoCaptureDevice, vs\_ScreenRecording, vs\_VideoFileOrURL, vs\_JPEGsOrBitmaps, vs\_VideoFromImages);

**enum** TVideoSource vs\_VideoCaptureDevice, vs\_ScreenRecording, vs\_VideoFileOrURL, vs\_JPEGsOrBitmaps, vs\_VideoFromImages ;

#### Description

#### Value Meaning

vs\_VideoCaptureDevice,  
vs\_ScreenRecording,  
vs\_VideoFileOrURL,  
vs\_JPEGsOrBitmaps,  
vs\_IPCamera,  
vs\_Mixer,  
vs\_VideoFromImages,  
vs\_ThirdPartyFilter,  
vs\_StreamInterface

These values are explained in the [VideoSource](#) property.

#### See Also

[Video sources supported for preview and recording](#) [VideoSource](#) [VideoSources](#) [VideoSourcesCount](#)

Created with the Standard Edition of HelpNDoc: [Make the switch to CHM with HelpNDoc's hassle-free WinHelp HLP to CHM conversion tool](#)

## TVMR9ImageAdjustment

### TVMR9ImageAdjustment

[Prev](#)
[Next](#)

Type of the VMR9 image adjustment settings.

#### Unit

[VidGrab](#)

#### Declaration

TVMR9ImageAdjustment=(vmr9\_Brightness, vmr9\_Contrast, vmr9\_Hue, vmr9\_Saturation, vmr9\_Alpha);

**enum** TVMR9ImageAdjustment vmr9\_Brightness, vmr9\_Contrast, vmr9\_Hue, vmr9\_Saturation, vmr9\_Alpha ;

#### Description

#### Value Meaning

vmr9\_Brightness  
vmr9\_Contrast  
vmr9\_Hue  
vmr9\_Saturation  
vmr9\_Alpha

#### See Also

[GetVMR9ImageAdjustmentBounds](#) [IsVMR9ImageAdjustmentAvailable](#) [SetVMR9ImageAdjustmentValue](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your Documentation Process with HelpNDoc's Project Analyzer](#)

## TVUMeter

### TVUMeter

[Prev](#)
[Next](#)



Type of the [VUMeter](#) property.

#### Declaration

TVuMeter=(vu\_Disabled, vu\_Analog, vu\_Bargraph);

**enum** TVuMeter vu\_Disabled, vu\_Analog, vu\_Bargraph ;

#### Description

##### Value Meaning

vu\_Disabled : VU-Meter disabled

vu\_Analog : analog VU-Meter

vu\_Bargraph : bargraph VU-Meter

vu\_AnalogOverlay: analog VU-meter overlayed over the video frames

vu\_BargraphOverlay: bargraph VU-meter overlayed over the video frames

#### See Also

[TVUMeterSetting](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#)  
[SpeakerBalance](#) [SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

Created with the Standard Edition of HelpNDoc: [Streamline Your CHM Help File Creation with HelpNDoc](#)

## TVUMeterSetting

### TVUMeterSetting

[Prev](#)

[Next](#)

Type used in the [SetVUMeterSetting](#) method.

#### Declaration

TVuMeter=(vu\_Handle, vu\_WarningPercent, vu\_PeakPercent, vu\_BkgndColor, vu\_NormalColor, vu\_WarningColor, vu\_PeakColor, vu\_TickSize, vu\_TickInterval, vu\_NeedleThickness);

**enum** TVuMeter vu\_Handle, vu\_WarningPercent, vu\_PeakPercent, vu\_BkgndColor, vu\_NormalColor, vu\_WarningColor, vu\_PeakColor, vu\_TickSize, vu\_TickInterval, vu\_NeedleThickness ;

#### Description

Type used in the [SetVUMeterSetting](#) method.

##### Value Meaning

vu_Handle	: sets the Handle of the panel or image control on which the VUMeter w
vu_WarningPercent	: percentage of the level above which the vu_WarningColor is use
vu_PeakPercent	: percentage of the level above which the vu_PeakColor is used
vu_BkgndColor	: color of the VU-meter background
vu_NormalColor	: color of the normal level (green by default)
vu_WarningColor	: color of the warning level (above the vu_WarningPercent)
vu_PeakColor	: color of the peak level (above the vu_PeakPercent)
vu_TickSize	: size of ticks in the bargraph VU-Meter
vu_TickInterval	: interval between ticks in the bargraph VU-Meter
vu_NeedleThickness	: thickness of the needle in the analog VU-Meter
vu_OverlayLeft	: left location of the VU-Meter when overlayed over the video fra
vu_OverlayTop	: top location of the VU-Meter when overlayed over the video fram
vu_OverlayWidth	: width of the VU-Meter when overlayed over the video frames
vu_OverlayHeight	: height of the VU-Meter when overlayed over the video frames
vu_Transparent(*)	: enables/disables the transparency of the background of the VU-M
vu_FlipVert(*)	: flips the VUMeter vertically when overlayed over the video fram
vu_FlipHorz(*)	: flips the VUMeter horizontally when overlayed over the video fr

vu\_CustomPercentValue : the VUmeter as a custom bargraph. Just pass (or update in real  
vu\_LogarithmicScale(\*) : the VU-meter scale is logarithmic (linear by default)

(\*) 0 = disabled, 1 = enabled

#### See Also

[TVUMeter](#) [AudioBalance](#) [AudioChannelRenderMode](#) [AudioVolume](#) [MuteAudioRendering](#) [SpeakerBalance](#)  
[SpeakerControl](#) [SpeakerVolume](#) [VUMeter](#)

---

Created with the Standard Edition of HelpNDoc: [Experience the Power and Ease of Use of a Help  
Authoring Tool](#)

---

## TWebcamStillCaptureButton

### TWebcamStillCaptureButton

[Prev](#)

[Next](#)

Type of the WebcamStillCaptureButton property.

#### Declaration

TWebcamStillCaptureButton=(wb\_Disabled, wb\_Enabled);

**enum** WebcamStillCaptureButton wb\_Disabled, wb\_Enabled ;

#### Description

Type of the [WebcamStillCaptureButton](#) property.

---

Created with the Standard Edition of HelpNDoc: [iPhone web sites made easy](#)

---