

# DNVideoX ActiveX control reference

AnMo Electronics Corporation

---

## Contents

License agreement .....	8
Notice .....	9
Tutorial .....	10
Create video preview .....	10
Capture still image.....	10
Capture video sequence .....	11
Upload video images to WEB server via FTP .....	12
Motion detection .....	12
Text captions on video (time-stamp, etc.) .....	13
Sending video frames through network.....	13
Upload files to WEB server via HTTP .....	14
Error codes reference.....	15
Constants.....	16
vcxUseDeinterlaceEnum.....	16
vcxUseVideoFilterEnum.....	16
vcxVideoRendererEnum .....	16
Properties .....	17
AudioCodecIndex .....	17
AudioDeviceIndex.....	17
AudioInputIndex.....	17
CapFilename.....	17
CapTimeLimit.....	17
CapTimeLimitEnabled.....	18
CaptureAudio .....	18
CaptureRate.....	18
ColorFormat .....	18
Connected .....	19
DebugMode.....	19

EnableNewFrameEvent .....	20
FTPPassiveMode.....	20
HalfSizedVideo.....	20
HasOverlay .....	20
hWnd.....	20
IsCapturing .....	20
LocalAddress.....	21
MasterStream.....	21
MouseIcon.....	21
MousePointer .....	21
Overlay .....	22
Overscan.....	22
Preview.....	22
PreviewAudio .....	22
PreviewFullScreen .....	23
PreviewScale.....	23
ProfileData.....	23
ProfileIndex .....	23
ServerMode.....	24
ServerPassword .....	24
ServerPort.....	24
ServerQuality.....	24
SyncUsingStreamOffset.....	24
UseDeinterlace .....	24
UseOverlay .....	25
UserFilter2CLSID .....	25
UserFilter3CLSID .....	25
UserFilterCLSID .....	25
UserFilterIUnknown .....	25
UseVideoFilter .....	26
Version.....	26
VideoCodecIndex.....	26
VideoCodecQuality.....	27
VideoDeviceIndex.....	27

VideoRotateAngle.....	27
VideoFlip.....	27
VideoHeight.....	27
VideoInputIndex.....	28
VideoProcAmp.....	28
VideoRenderer.....	29
VideoSourceURL.....	29
VideoWidth.....	29
WMAttributes.....	29
WMTVersion.....	30
Methods.....	31
AboutBox.....	31
AllocCapFile.....	31
BarcodeInit.....	31
GetColorIn.....	31
EnableMicroTouch.....	31
FreezePreview.....	31
CompareImages.....	32
CopyCaptureFile.....	32
CopyFrame.....	32
DetectMotion.....	32
DisplayRemote.....	33
FreezeAWB.....	33
GetActualFrameRate.....	33
GetAMR.....	33
GetAudioCodecCount.....	33
GetAudioCodecName.....	34
GetAudioDeviceCount.....	34
GetAudioDeviceName.....	34
GetAudioFormat.....	34
GetAudioInputCount.....	34
GetAudioInputName.....	34
GetAudioLevel.....	34
GetAudioLevel2.....	35

GetAutoExposure .....	35
GetAWBB .....	35
GetAWBG .....	35
GetAWBR .....	35
GetCapFileSize .....	36
GetCapStatus .....	36
GetDateCode .....	36
GetDeviceID .....	36
GetAETarget .....	37
GetAESTability .....	37
GetBarcode .....	37
GetExposureValue .....	37
GetGain .....	38
GetFilterSettings .....	38
GetFrameAsHBITMAP .....	38
GetLEDState .....	38
GetLuma .....	39
GetMirror .....	39
GetProfileCount .....	39
GetProfileDesc .....	39
GetProfileName .....	39
GetRGB .....	40
GetSobelCenter .....	40
GetTimecode .....	40
GetVideoCaps .....	40
GetVideoCodecCount .....	41
GetVideoCodecName .....	41
GetVideoDeviceCount .....	41
GetVideoDeviceDesc .....	41
GetVideoDeviceName .....	41
GetVideoFormat .....	42
GetVideoInputCount .....	42
GetVideoInputName .....	42
GetVideoProcAmpValueRange .....	42

GetVMR9IUnknown.....	43
GrabFrame.....	43
HTTPUpload.....	43
KnobMotorRotate .....	44
LoadProfileFromURL.....	44
PauseCapture .....	44
PlayRemoteAudio.....	44
ReceiveAudio.....	45
ReceiveFrame .....	45
Recompress .....	45
RecompressEx .....	45
ResumeCapture.....	46
SaveEDR.....	46
SaveEDOF .....	46
SaveFrame.....	46
SaveFrameJPG .....	46
SavePictureJPG.....	47
SendScriptCommand.....	47
SetAudioDelay .....	48
SetAudioFormat .....	48
SetAudioInputLevel .....	50
SetAudioVolume.....	50
SetAutoExposure.....	50
SetAWBB.....	50
SetAWBG .....	51
SetAWBR.....	51
SetBitmapOverlay.....	51
SetChromaKey .....	51
SetCrop.....	52
SetAETarget .....	52
SetAESTability .....	52
SetExposureValue.....	52
SetExposureTime.....	53
SetFLCSwitch .....	53

SetFLCLevel.....	54
SetGain .....	54
SetFadeLevel.....	54
SetFilterSettings .....	55
SetFlicker .....	55
SetHighPriority .....	55
SetLEDState .....	55
SetLEDStrobeLength.....	56
SetMasterAudioVolume .....	56
SetMirror .....	57
SetMotionMask .....	57
SetTextOverlay .....	57
SetVideoFormat.....	58
SetVideoFormatEx.....	58
SetZoom .....	58
ShowAudioCodecDlg .....	58
ShowAudioFormatDlg .....	59
ShowAudioSourceDlg.....	59
ShowBarcodeSettingDlg.....	59
ShowUserFilterDlg.....	59
ShowVideoCodecDlg .....	59
ShowVideoCrossbarDlg .....	60
ShowVideoFormatDlg.....	60
ShowVideoSourceDlg .....	60
SingleFrameAdd .....	60
SingleFrameAddPicture .....	60
SingleFrameClose .....	61
SingleFrameOpen .....	61
StartBroadcast.....	61
StartBroadcastPush .....	62
StartCapture .....	62
StopBroadcast .....	63
StopCapture.....	63
UploadFile.....	63

UploadFrame .....	63
CaptureEnd .....	63
CaptureReady .....	64
CaptureStart .....	64
DeviceLost .....	64
FootPedalPressed .....	64
FullscreenLost .....	64
NewFrame .....	64
RecompressCompleted .....	64
RecompressProgress .....	65
MicroTouchPressed .....	65
Pix2Length .....	65
Pix2Length2 .....	65

## License agreement

This Limited Use Software License Agreement (the "Agreement") is a legal agreement between you, the end-user ("Licensee"), and AnMo Electronics Corp ("AnMo"). By using or storing this program ("DNVideoX") on a computer hard drive or other media, you are agreeing to be bound by the terms of this Agreement.

Licensee may not alter this DNVideoX in any way, including changing or removing any messages or windows.

Licensee may not decompile, reverse engineer, disassemble or otherwise reduce this DNVideoX to a human perceivable form. Licensee may not modify, rent or resell the DNVideoX for profit. Licensee may not publicize or distribute any registration code algorithms, information, or registration codes used by the DNVideoX without permission of AnMo.

Licensee written applications containing embedded DNVideoX control may be freely distributed, without royalty payments to AnMo, provided that such distributed product is bound into these applications in such a way so as to prohibit separate use in design mode, and that such product is distributed only in conjunction with the hardware manufactured by AnMo.

This DNVideoX may be used as a constituent control only if the compound control thus created is distributed with and as an integral part of an application. The license may not be transferred to a third party under any circumstance.

This DNVideoX is provided by AnMo on an "as is" basis. AnMo makes no warranty, expressed or implied, including without limitation the implied warranties of non-infringement, merchantability and fitness for a particular purpose, regarding the DNVideoX or its use and operation alone or in combination with any product. Under no circumstances shall AnMo be liable for any incidental or consequential damages, nor for any damages in excess of the original purchase price.



## Notice

1. Please be sure the SDK is installed with Administrator's right.

2. The following files need to be included in your application setup:

ClientPropertyPageLIB.dll, SMIUtility.dll, d3dx9\_31.dll, and DNVideoX.ocx are located in "DNVideoX SDK" program folder, and need to be copied to Windows system folder on target PC.

The DNVideoX.ocx must be registered as any other ActiveX control with regsvr32.exe. For user who has registered the prior version of DNVideoX.ocx, please unregister it for replacing the existing DNVideoX.ocx with the updated one.

DNLBarReader.dll, enfuse.exe, Microsoft.VC90.CRT.manifest, msvcm90.dll, msvcp90.dll and msvcr90.dll are also located in "DNVideoX SDK" program folder, and need to be copied to your application folder.

3. Some built-in or external webcams may affect functionality controlling the sensor or LED. It is suggested to disable or unplug the webcam before using the program developed with the SDK.

## Tutorial

### Create video preview

To have DNVideoX showing video preview, just set Connected and Preview properties to True.

#### *Visual Basic example:*

Put DNVideoX control onto empty form and paste this code into "Declarations" section;

```
Private Sub Form_Load()  
    'Connect control to video driver  
    DNVideoX1.Connected = True  
    'Start preview  
    DNVideoX1.Preview = True  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    'Disconnect control from video capture driver  
    DNVideoX1.Connected = False  
End Sub
```

---

### Capture still image

```
Picture1.Picture=DNVideoX.GrabFrame
```

#### *Save image to JPG file*

DNVideoX.SaveFrameJPG "mypicture.jpg",90,1

Connected property must be set to TRUE for these methods to work.

---

## Capture video sequence

### *AVI capture:*

To capture video into AVI file, use StartCapture method. StopCapture stops recording.

Specify output AVI file in CapFilename property.

Example:

```
DNVideoX.Connected=True  
DNVideoX.Preview=True  
DNVideoX.CapFilename = "c:\movie.avi"  
DNVideoX.StartCapture  
  
'wait for user click to finish  
MessageBox "Press OK to stop capture!"  
  
DNVideoX.StopCapture
```

### *WMV capture:*

CapFilename must have extension ".WMV" .

Before starting WMV capture, set [ProfileIndex](#) property.

See also: [CapTimeLimit](#), [CapTimeLimitEnabled](#), [CapFilename](#), [CaptureAudio](#)

For WMV capture, see also: [WMTVersion](#), [ProfileIndex](#), [ProfileData](#), [GetProfileCount](#), [GetProfileName](#), [ProfileData](#)

Upload video images to WEB server via FTP

Use [UploadFrame](#) method.

Example

```
DNVideoX.UploadFrame("ftp.foo.com","john","tiger","images","mypic.jpg",21,70)
```

See also [FTPPassiveMode](#) property.

---

Motion detection

[DetectMotion](#) method Return the number between 0 and 100, reflecting the change detected in front of the camera.

Call this method every second or two and check the result it Return. If the result is greater than 30, there is something moving in front of the camera. Experiment with this to see which value to use.

**Example**

```
Sub Timer1_OnTimer()
```

```
If DNVideoX1.DetectMotion>30 then
```

```
    Beep
```

```
    MsgBox "Where are you going?"
```

```
End If
```

```
End Sub
```

## Text captions on video (time-stamp, etc.)

To set date-time stamp in top-left corner on video, use:

DNVideoX.[SetTextOverlay](#) 0, "TIME", 0, 0, "Arial", 14, RGB(255,0,0), -1

---

## Sending video frames through network

See also: [Upload image to web server using FTP](#)

### Properties:

<a href="#">ServerMode</a>	Set this to TRUE to accept connections
<a href="#">ServerPort</a>	Specify IP port number for connection
<a href="#">ServerPassword</a>	Specify password for received frames

### Methods:

<a href="#">ReceiveFrame</a>	Connect and get frame from remote server
------------------------------	--

### Events:

<a href="#">ConnectionRequest</a>	This is invoked on a server when client connects
-----------------------------------	--

At server side, use these steps:

1. connect the camera

2. start preview
3. set ServerMode property of DNVideoX to True

And, at the client side, use:

```
PictureBox1.Picture = DNVideoX1.ReceiveFrame (serveraddress )
```

See "Network video transfer" sample in DNVideoX samples...

[Upload files to WEB server via HTTP](#)

See [HTTPUpload](#) method topic.

## Error codes reference

Error code	Error message	Method
1001	Error creating MPEG demux filter	StartCapture
1002	Not supported in overlay mode	GrabFrame
1003	Error in bitmap buffer filter	GrabFrame
1004	Exception occurred	DetectMotion
1005	Picture is not a bitmap	SavePictureJPG
1006	UseVideoFilter property must be set to TRUE	ServerMode
1007	Video driver not connected	ShowVideoCodecDlg
1008	Video codec property not set	ShowVideoCodecDlg
1009	Audio codec property not set	ShowAudioCodecDlg
1010	Can't open capture while video resolution is unknown. Start preview first	SingleFrameOpen
1011	Can't find WMVCORE.DLL	GetProfileCount
1012	Can't find new version of WMVCORE.DLL	GetProfileCount
1013	Can't find WMVCORE.DLL	StartBroadcast
1014	Can't find new version of WMVCORE.DLL	StartBroadcast
1015	Exception occurred	StartCapture2
1016	Connected must be set to TRUE	GetAudioInputCount
1017	Can't find WMVCORE.DLL	StartBroadcast
1018	Can't find new version of WMVCORE.DLL	StartBroadcast
1019	Picture must be an icon	SetMouseIcon
1020	Picture is not a bitmap	SetChromaKey
1021	Error creating bitmap	GrabFrame
1022	Error while copying bitmap	GrabFrame
1023	Error while getting bitmap	GrabFrame
1024	GrabFrame failed	SaveFrameJPG
1025	Invalid bitmap	SaveFrameJPG
1026	Exception occurred	SaveFrameJPG
1027	Can't open JPEG file	SaveFrameJPG
1028	JPEG creation error	SaveFrameJPG
1029	Couldn't get image from the URL	Connected
1030	Trial period expired	Connected
1031	No video hardware detected or error while connecting to device	Connected
1032	Can't build preview graph	Preview
1033	Preview failed	Preview
1034	Error creating MPEG demux filter	Preview
1035	Audio format must have 16 bits per sample	Preview=True
1036	Audio format must be PCM	Preview=True
2000	Can't run graph	StartCapture

## Constants

### vcxUseDeinterlaceEnum

vcxNone	0
vcxSimple	1
vcxBob	2

### vcxUseVideoFilterEnum

vcxNo	0
vcxBoth	1
vcxPreviewOnly	2
vcxGrabOnly	3

### vcxVideoRendererEnum

vcxSystemDefaultRenderer	0
vcxVMR9	1
vcxVMR7	2
vcxGDI	3



## Properties

### AudioCodecIndex

Set index of audio codec to use for audio compression

```
Property AudioCodecIndex As Long
```

### AudioDeviceIndex

Set index of audio device to use for capturing audio

```
Property AudioDeviceIndex As Long
```

If this property is set to value of 100 and CaptureAudio is TRUE, DNVideoX uses video device for audio capturing also.

### AudioInputIndex

Specify input port for audio on multi-port audio input cards

```
Property AudioInputIndex As Long
```

### CapFilename

Filename for captured media file. Extension can be AVI or WMV.

```
Property CapFilename As String
```

The default capture filename is "c:\capture.avi".

To split captured movie into several files, just change this property to a new filename while capture is running.

If capture filename ends with ".WMV", WindowsMedia format is used.

### CapTimeLimit

Time limit for capturing, in seconds

```
Property CapTimeLimit As Long
```

The default value is 60.

This property will work only if CapTimeLimitEnabled is set to TRUE.

### CapTimeLimitEnabled

Indicate is CapTimeLimit property valid

```
Property CapTimeLimitEnabled As Boolean
```

The default value for this property is FALSE.

If this value is TRUE video capture will stop after CapTimeLimit seconds.

### CaptureAudio

Indicate will audio be captured

```
Property CaptureAudio As Boolean
```

The default value is TRUE.

If this value is FALSE, captured video will have no sound.

### CaptureRate

Get/Set video capture rate (number of frames per second)

```
Property CaptureRate As Double
```

The default value is 15. CaptureRate property, to have an effect, must be set before connecting capture driver.

### ColorFormat

Specify color format of the source video stream

```
Property ColorFormat As Long
```

Values are:

- 0 = RGB24 (default),
- 1 = CLPL,
- 2 = YUYV,
- 3 = IYUV,

- 4 = YVU9,
- 5 = Y411,
- 6 = Y41P,
- 7 = YUY2,
- 8 = YVYU,
- 9 = UYVY,
- 10 = Y211,
- 11 = YV12,
- 12 = CLJR,
- 13 = IF09,
- 14 = CPLA,
- 15 = MJPG,
- 16 = TVMJ,
- 17 = WAKE,
- 18 = CFCC,
- 19 = IJPG,
- 20 = Plum,
- 21 = DVCS,
- 22 = DVSD,
- 23 = MDVF,
- 24 = RGB1,
- 25 = RGB4,
- 26 = RGB8,
- 27 = RGB565,
- 28 = RGB555,
- 29 = RGB32

DNVideoX.Connected = True

DNVideoX.ColorFormat = 3

DNVideoX.StartCapture

### Connected

Get/set connection to video device

Property Connected As Boolean

Connects DNVideoX control onto video capture device specified with VideoDeviceToUse property.

DNVideoX.Connected = True

### DebugMode

Internal. Do not use.

```
Property DebugMode As Long
```

### EnableNewFrameEvent

If set to TRUE, every new video frame will generate NewFrame event

```
Property EnableNewFrameEvent As Boolean
```

### FTPPassiveMode

If set to TRUE, FTP transfer methods will use passive mode.

```
Property FTPPassiveMode As Boolean
```

Passive FTP mode is firewall / proxy friendly.

### HalfSizedVideo

If set to TRUE, DNVideoX will resize video to half of it's actual size when capturing

```
Property HalfSizedVideo As Boolean
```

### HasOverlay

(Read-only) Return TRUE if selected video device supportes video overlay feature

```
Property HasOverlay As Boolean
```

### hWnd

Return Windows window handle of DNVideoX control

```
Property hWnd As Long
```

### IsCapturing

Return TRUE if video-capture is in progress

```
Property IsCapturing As Boolean
```

### LocalAddress

If multiple network adapters are installed, this property specifies which one to use in network communication.

```
Property LocalAddress As String
```

### MasterStream

Specify master stream in AVI file (audio or video or none)

```
Property MasterStream As Long
```

Values are:

- 1 = none
- 0 = video
- 1 = audio (default)

### MouseIcon

Set custom mouse icon

```
Property MouseIcon As stdole.Picture
```

### MousePointer

Set mouse pointer shape

```
Property MousePointer As Long
```

Mouse pointer values are:

- 0 (Default) Shape determined by the object.
- 1 Arrow.
- 2 Cross (crosshair pointer).
- 3 I beam.
- 4 Icon (small square within a square).
- 5 Size (four-pointed arrow pointing north, south, east, and west).
- 6 Size NE SW (double arrow pointing northeast and southwest).
- 7 Size N S (double arrow pointing north and south).
- 8 Size NW SE (double arrow pointing northwest and southeast).
- 9 Size W E (double arrow pointing west and east).
- 10 Up Arrow.
- 11 Hourglass (wait).
- 12 No Drop.
- 13 Arrow and hourglass.

- 14 Arrow and question mark.
- 15 Size all.

## Overlay

Enable/disable video overlay preview

`Property Overlay As Boolean`

If HasOverlay property is TRUE, Overlay can be used instead of Preview property to show live video on-screen. Overlay video doesn't travel through main memory and it consumes almost no CPU resources. However, frame grabbing doesn't work. To be able to capture still frames and put text/bitmap over the video, you must use Preview property. Connected property must be set to TRUE before you can use this property.

```
DNVideoX.Connected = True
DNVideoX.Overlay = True
```

## Overscan

Specify how many pixels to discard at video borders.

`Property Overscan As Long`

## Preview

Enable/disable video preview

`Property Preview As Boolean`

If set to True, starts video preview in DNVideoX window. Must be connected to use this property.

```
DNVideoX.Preview = True
```

## PreviewAudio

Set this property to TRUE if you want audio in preview mode

`Property PreviewAudio As Boolean`

This property should be set before Preview=True.

### PreviewFullScreen

When set to TRUE, preview video will cover the whole screen

```
Property PreviewFullScreen As Boolean
```

PreviewFullScreen property should be set to TRUE after form (window) with DNVideoX is shown on screen.

### PreviewScale

Indicate will preview video be resized to fit control rectangle

```
Property PreviewScale As Boolean
```

The default value is TRUE. If this is FALSE, image in preview window will be exact size as captured, and if it is TRUE, image will stretch to fit control size on screen.

### ProfileData

Set custom WM profile XML data

```
Property ProfileData As String
```

Custom profiles for WMV capture (for example: high quality video stream of 10Mbps) can be created with Windows Media Profile Editor tool included with MS Media Encoder9 . Custom profiles are saved into .prx files. PRX files are in plain-text XML format. You can copy XML data to ProfileData property to use this custom profile while capturing to WMV file.

### ProfileIndex

Specify system profile to use when creating WMV files

```
Property ProfileIndex As Long
```

A profile is a collection of data that describes the configuration of an WMV file. The stream information in a profile contains the bit rate, buffer window, and media properties for the stream. The stream information for audio and video describes exactly how the media is configured in the file, including which codec (if any) will be used to compress the data.

You can create custom profiles with ProfileEdit tool of WMFormat SDK. To use custom profile, set ProfileData property.

Use GetProfileCount, GetProfileName and GetProfileDesc methods to enumerate system profiles.

```
DNVideoX.Connected=True  
DNVideoX.Preview=True  
DNVideoX.CapFilename="capture.wmv"
```

DNVideoX.ProfileIndex=26

DNVideoX.StartCapture

### ServerMode

If set to TRUE, the control will listed for TCP connections on ServerPort mode and send video frames

```
Property ServerMode As Boolean
```

If set to TRUE, the control will listed for TCP connections on ServerPort mode and send video frames

### ServerPassword

Set password for server access

```
Property ServerPassword As String
```

### ServerPort

Number of TCP port for ServerMode

```
Property ServerPort As Long
```

### ServerQuality

Specify quality of video images transfered by ReceiveFrame method. Range 10-100. Default 30.

```
Property ServerQuality As Long
```

### SyncUsingStreamOffset

Is stream offset used to synchronize audio/video streams in captured file

```
Property SyncUsingStreamOffset As Boolean
```

### UseDeinterlace

Deinterlace video in DNVideoX video-filter

```
Property UseDeinterlace As vcxUseDeinterlaceEnum
```

Possible values are:



vcxNone=0 - no deinterlacing  
vcxSimple=1 - simple (fast) deinterlacing  
vcxBob=2 - Better deinterlacing

UseVideoFilter must be enabled for deinterlacing to work.

### UseOverlay

If set to TRUE, DNVideoX will output video preview to overlay surface of graphics adapter.

```
Property UseOverlay As Boolean
```

### UserFilter2CLSID

Second user filter

```
Property UserFilter2CLSID As String
```

### UserFilter3CLSID

Third user filter

```
Property UserFilter3CLSID As String
```

### UserFilterCLSID

Specify user video filter by CLSID

```
Property UserFilterCLSID As String
```

Set this property to CLSID string of custom video filter you want to use. It should be set before setting Connected to TRUE.

DNVideoX will create instance of this CLSID and release it after Connected is set to FALSE.

```
DNVideoX.UserFilterCLSID = "{c200e360-38c5-11ce-ae62-08002b2b79ef}"
```

```
DNVideoX.Connected = TRUE
```

```
DNVideoX.Preview = TRUE
```

### UserFilterIUnknown

Set IUnknown pointer of custom video filter.

**Property UserFilterIUnknown As Unknown**

You must create the filter, set its properties and QueryInterface for IUnknown. Then, set this property to your filter IUnknown pointer before setting Connected property of DNVideoX to TRUE. DNVideoX doesn't call AddRef or Release on this interface. It will QueryInterface for IBaseFilter and call AddRef/Release on it.

**UseVideoFilter**

Determine if DNVideoX video filter will be used. This filter handles frame grabbing, video cropping and text/bitmap overlay. Without it, video stream can be much faster.

**Property UseVideoFilter As vcxUseVideoFilterEnum**

Possible values are:

vcxNo=0,

vcxBoth=1,

vcxPreviewOnly=2,

vcxGrabOnly=3

"vcxNo" means fastest preview and capture,

"vcxGrabOnly" is little to nothing slower but offer still video frame grabbing capability (with GrabFrame method) without text/image overlay,

"vcxPreviewOnly" can be used when you want to put text/image overlay on video preview while capturing unmodified video,

"vcxBoth" means that both video preview and captured video will have text/image overlays.

DNVideoX.UseVideoFilter = vcxNo

DNVideoX.Connected = True

DNVideoX.Preview = True

**Version**

Return DNVideoX.OCX version number

**Property Version As String****VideoCodecIndex**

Set index of video codec to use for on-the-fly compression of video

**Property VideoCodecIndex As Long**

### VideoCodecQuality

Set quality parameter for video codec

```
Property VideoCodecQuality As Long
```

### VideoDeviceIndex

Set index of video device to use for capture

```
Property VideoDeviceIndex As Long
```

### VideoRotateAngle

Set video rotation in degrees. Angle value can be in range from -180 to 180.

```
Property VideoRotateAngle As Long
```

### VideoFlip

Set video flipping. Flips video image horizontally and/or vertically.

```
Property VideoFlip As Long
```

VideoFlip values are:

0 = no flipping (default)

1 = horizontal flip

2 = vertical flip

3 = horizontal+vertical flip

If the video capture device doesn't support video flipping, you can use DNVideoX video filter to manually flip the image. Just use -1, -2 or -3 as values for this property.

If VideoFlip property Return -1, it indicates that Connected property should be set to TRUE. If VideoFlip value is -2, it means that video flipping isn't supported by current video device.

DNVideoX.Connected = True

DNVideoX.Preview = True

DNVideoX.VideoFlip = 1

### VideoHeight

Return current video height in pixels. This property is read-only.

`Property VideoHeight As Long`

### VideoInputIndex

Set channel to use on multi-port capture cards

`Property VideoInputIndex As Long`

### VideoProcAmp

Get/Set video properties

`Property VideoProcAmp (ValueIndex As Long) As Long`

ValueIndex parameter is:

- Brightness = 0,
- Contrast = 1,
- Hue = 2,
- Saturation = 3,
- Sharpness = 4,
- Gamma = 5,
- ColorEnable = 6,
- WhiteBalance = 7,
- BacklightCompensation = 8,
- Gain = 9

DNVideoX offers internal video-settings filter. You can use it by specifying following values as ValueIndex parameter and value range in parenthesis:

- 100 = Brightness (-255 to 255)
- 101 = Contrast (-100 to 100)
- 102 = Hue (-180 to 180)
- 103 = Saturation (-100 to 412)
- 105 = Gamma (1 to 500)
  
- 100 = Add Red value to pixel color
- 101 = Add green value to pixel color
- 102 = Add blue value to pixel color
- 103 = Blue mode
- 104 = B/W mode
- 105 = Red mode
- 106 = Binary mode treshold
- 107 = Binary mode RGB color value

## Dim Brightness As Long

Brightness = DNVideoX.VideoProcAmp (0)

Dim MinVal As Long ,MaxVal As Long ,StepD As Long ,DefVal As Long  
DNVideoX.GetVideoProcAmpValueRange 0, MinVal, MaxVal, StepD, DefVal

'set brightness here

Brightness=MinVal+((MaxVal-MinVal)/2)

DNVideoX.VideoProcAmp(0)=Brightness

## VideoRenderer

Select video renderer to use for video preview.

`Property VideoRenderer As vcxVideoRendererEnum`

0 = vcxSystemDefault: Use default DirectShow video renderer,

1 = vcxVMR9: Use new Direct3D VMR9 renderer

2 = vcxVMR7: use older VMR7 DirectDraw-based renderer

3 = vcxGDI: use GDI-based video renderer

## VideoSourceURL

URL of network camera acting as video source

`Property VideoSourceURL As String`

VideoDeviceIndex property must be set to -2 for this property to work.

vcx.VideoDeviceIndex = -2

vcx.VideoSourceURL = "http://mycam/image.jpg"

vcx.AudioDeviceIndex = -1

vcx.Connected = True

vcx.Preview = True

## VideoWidth

Return current video width in pixels. This property is read-only.

`Property VideoWidth As Long`

## WMAttributes

Windows Media attributes to set when capturing into WMV files or broadcasting

**Property WMAAttributes As String**

WMAAttributes property is a string delimited with '|' with these fields: Title, Author, Copyright, Rating, Description . It can be an empty string if no attributes are needed.

vcx.WMAAttributes="my title|author is me|copyright to me|rating is 5|this is description "

**WMTVersion**

Set WindowsMedia system profiles version to use. See ProfileIndex property. Deafult is 7. Possible values are 4,7 and 8.

**Property WMTVersion As Long**

## Methods

### AboutBox

Shows About Box of DNVideoX

```
Sub AboutBox
```

### AllocCapFile

Pre-allocates space on disk for capture file

```
Function AllocCapFile (FileSizeMb As Long) As Long
```

You can improve streaming capture performance significantly by preallocating a capture file large enough to store an entire video clip and by defragmenting the capture file before capturing the clip.

### BarcodeInit

Enable the barcode reading capability.

```
Function BarodeInit ()
```

### GetColorIn

Return average RGB color value in specified rectangle on video.

```
Function GetColorIn (X As Long, Y As Long, W As Long, H As Long) As Long
```

### EnableMicroTouch

Enable/disable MicroTouch button on the camera. When enabled, pressing the button will trigger MicroTouchPressed event.

```
Function EnaleMicroTouch (Enable As Boolean) As Boolean
```

### FreezePreview

Pause video preview.

```
Function FreezePreview (Freeze As Long) As Long
```

Freeze parameter is 1 for pause, 0 for resume.

## CompareImages

Return difference image between two images.

```
Function CompareImages (Picture1 As stdole.Picture, Picture2 As stdole.Picture,  
Treshold As Long, Color1 As Long, Color2 As Long) As stdole.Picture
```

This method Return a binary color image of different pixels in two images.

Picture1.Picture = DNVideoX.GrabFrame

-or-

Picture1.Picture = DNVideoX.ReceiveFrame ( serveraddr )

Picture3.Picture = DNVideoX.CompareImages(Picture1, Picture2, 25, RGB(0,0,0), RGB(255,255,255))

## CopyCaptureFile

Copies AVI file from pre-allocated storage into new file

```
Function CopyCaptureFile (New As String) As Long
```

## CopyFrame

Copy current vide frame into clipboard

```
Function CopyFrame As Boolean
```

none

## DetectMotion

Detect changes in video frames

```
Function DetectMotion As Long
```

The DetectMotion method Return the number between 0 and 100, reflecting the change detected in front of the camera.

Call this method every second or two and check the result it Return. If the result is greater than 30, there is something moving in front of the camera. Experiment with this to see which value to use.

See SetMotionMask method.

Sub Timer1\_OnTimer()



```
If DNVideoX1.DetectMotion>30 then  
Beep  
MsgBox "Where are you going?"  
End If  
End Sub
```

### DisplayRemote

Starts a video-conference call

```
Function DisplayRemote (RemoteAddress As String, Audio As Boolean) As Long
```

```
'call a computer with IP address of 192.168.0.5  
clientvcx.DisplayRemote "192.168.0.5", True
```

```
'or, with a network address as 'johnsoffice'  
clientvcx.DisplayRemote "johnsoffice", True
```

### FreezeAWB

Freeze or unfreeze the AWB. It is needed before using SetAWBR, SetAWBG, and SetAWBB. This function is only available for 5M Premier and 5M Edge series.

```
Function FreezeAWB (DeviceIndex As Long, FreezeAWBEnable As Long)
```

FreezeAWBEnable argument values:

- 0 : AWB freeze disabled
- 1 : AWB freeze enabled

On error, value of -1 is returned.

### GetActualFrameRate

Return current actual frame rate. Some devices may provide lower frame rates than requested because of bandwidth availability. This is only available during video streaming.

```
Function GetActualFrameRate As Double
```

### GetAMR

Return the magnification reading from models with AMR capability such as AM4515 series.

```
Function GetAMR (DeviceIndex As Long) As Double
```

### GetAudioCodecCount

Return installed audio codec count

```
Function GetAudioCodecCount As Long
```

### GetAudioCodecName

Return audio codec name

```
Function GetAudioCodecName (Index As Long) As String
```

### GetAudioDeviceCount

Return number of audio devices in the system

```
Function GetAudioDeviceCount As Long
```

### GetAudioDeviceName

Return audio device name

```
Function GetAudioDeviceName (Index As Long) As String
```

### GetAudioFormat

Return audio format parameters

```
Function GetAudioFormat (FmtTag As Long, nChannels As Long, nSamplesPerSec As Long,  
nAvgBytesPerSec As Long, nBlockAlign As Long, wBitsPerSample As Long) As Boolean
```

### GetAudioInputCount

Return number of input ports on audio source

```
Function GetAudioInputCount As Long
```

### GetAudioInputName

Return audio input port name

```
Function GetAudioInputName (Index As Long) As String
```

### GetAudioLevel

Return audio level in preview mode

```
Function GetAudioLevel As Long
```

### GetAudioLevel2

Get audio levels for left and right channel

```
Function GetAudioLevel2 (left As Long, right As Long) As Long
```

### GetAutoExposure

Return AutoExposure of current video frame.

```
Function GetAutoExposure (DeviceIndex As Long) As Long
```

AutoExposure value:

0=AE off

1=AE on

### GetAWBB

Return AWB Blue setting of current video frame.

\* This method is only available for Premier and Edge series.

```
Function GetAWBB (DeviceIndex As Long) As Long
```

Return value is in range 0 to 255.

### GetAWBG

Return AWB Green setting of current video frame.

\* This method is only available for Premier and Edge series excepting AM4011 and AM4013.

```
Function GetAWBG (DeviceIndex As Long) As Long
```

Return value is in range 0 to 255.

### GetAWBR

Return AWB Red setting of current video frame.

\* This method is only available for Premier and Edge series.

```
Function GetAWBR (DeviceIndex As Long) As Long
```

Return value is in range 0 to 255.

### GetCapFileSize

Return file size (in bytes) of capture file

```
Function GetCapFileSize As Double
```

On error, value of -1 is returned.

By using this method, you can check capture file size while capturing is in progress. That way, it gives you a chance to stop capture, change CapFilename and Startcapture again, if you want to keep capture files under some size limit.

### GetCapStatus

Retrieves video capture parameters

```
Function GetCapStatus (ImageWidth As Long, ImageHeight As Long, CurrentVideoFrame  
As Long, CurrentVideoFramesDropped As Long, CurrentTimeElapsedMS As Long,  
fCapturingNow As Long) As Long
```

Return TRUE if successful or FALSE if window is not connected to a capture driver.

This method Return various status information in variables passed to it as arguments. ImageHeihgt and Width are in pixels, fCapturingNow is TRUE is capture is in progress.

### GetDateCode

Return time on DV video tape (the time when the video has been taken)

```
Function GetDateCode As Date
```

See [GetTimecode](#) method.

### GetDeviceID

Return unique device ID string

```
Function GetDeviceID (DeviceIndex As Long) As String
```

If DeviceIndex parameter is in range 0-1000, video device ID is returned. If DeviceIndex is in range

1000-2000, audio device ID is returned (first audio device is 1000, second 1001, etc)

### GetAETarget

Return Auto-Exposure target value of current video frame.

```
Function GetAETarget (DeviceIndex As Long) As Long
```

### GetAESTability

Return Auto-Exposure's Stability value of current video frame.

\* This method is only available for 1.3M Premier and 1.3M Edge series.

```
Function GetAESTability (DeviceIndex As Long) As Long
```

### GetBarcode

Return the barcode if detected. To use GetBarcode, the barcode reading functionality needs to be enabled by BarcodeInit method.

```
Function GetBarcode () As Variant
```

```
Dim aa As Variant  
BarCodeReturn = v.GetBarcode  
List1.Clear  
VarType (BarCodeReturn)  
If UBound(BarCodeReturn) >= LBound(BarCodeReturn) Then  
    For f = LBound(BarCodeReturn) To UBound(BarCodeReturn)  
        List1.AddItem BarCodeReturn(f, 0) 'BarCodeString  
        List1.AddItem BarCodeReturn(f, 1) 'BarCode Type  
        List1.AddItem BarCodeReturn(f, 2) 'BarCode Orientation.  
        List1.AddItem BarCodeReturn(f, 3) 'BarCode Location  
    Next f  
End If
```

### GetExposureValue

Return the relative Exposure value, which is proportional to the exposure time, of current video frame.

\* This method is only available for Premier and Edge series.

```
Function GetExposureValue (DeviceIndex As Long) As Long
```

### GetGain

Return Gain of current video frame.

\* This method is only available for Premier and Edge series.

```
Function GetGain (DeviceIndex As Long) As Long
```

### GetFilterSettings

Return current filter settings

```
Function GetFilterSettings (FilterID As Long) As String
```

FilterID values:

- 1 = video compress filter
- 2 = audio compress filter
- 3 = video source filter
- 4 = audio source filter
- 5 = user filter

This method Return current settings of the filter as string. To set filter, call SetFilterSettings method.

If returned value is an empty string, selected filter doesn't support settings retrieval by code.

Dim s As String

DNVideoX.VideoCodecIndex = 6

DNVideoX.ShowVideoCodecDlg

s=DNVideoX.GetFilterSettings(1)

### GetFrameAsHBITMAP

Return Windows HBITMAP value of current video frame.

```
Function GetFrameAsHBITMAP As Long
```

### GetLEDState

Get the camera LED state.

\* This method may not applicable to AM211, AM2011, and Dino-Eye series.

```
Function GetLEDState(DeviceIndex As Long, LEDState As Long) As Long
```

LEDState Values:

0 = LED off

1 = LED1 on

2 = LED2 on. The LED2 only exists on models with 2 switchable LEDs.  
-1 = access error

### GetLuma

Return average Luma value of current video frame.

\* This method is only available for Premier and Edge series.

```
Function GetLuma (DeviceIndex As Long) As Long
```

Return value is in range from 0 to 255.

### GetMirror

Return Mirror value of current video frame.

\* This method is only available for Premier and Edge series.

```
Function GetMirror (DeviceIndex As Long) As Long
```

Mirror value:

0=normal

1=vertical mirror

2=horizontal mirror

3=vertical +horizontal mirror

### GetProfileCount

Return number of WindowsMedia system profiles

```
Function GetProfileCount As Long
```

See WMTVersion property.

### GetProfileDesc

Return WindowsMedia profile description

```
Function GetProfileDesc (ProfileIndex As Long) As String
```

### GetProfileName

Return WindowsMedia profile name

```
Function GetProfileName (ProfileIndex As Long) As String
```

### GetRGB

Return current video frame image as array of RGB values.

```
Function GetRGB
```

```

Dim a() As Byte
a = DNVideoX1.GetRGB
For y = 0 To 239 'image height is 240 pixels in this case
For x = 0 To 319 '320 pixels
i = (y * (320 * 3)) + (x * 3)
'NOTE: byte order isn't RGB, it's BGR
PSet (x, y), RGB(a(i + 2), a(i + 1), a(i))
Next x
Next y

```

### GetSobelCenter

Return Sobel score of image center (320x240 pixels) for indicating the sharpness of edges.

```
Function GetSobelCenter (DeviceIndex As Long)
```

### GetTimecode

Return timecode value on digital VCR video type

```
Function GetTimecode As Long
```

Returned value is:

Hours, minutes, seconds, and frames, as a binary coded decimal (BCD) value: 0xhhmmssff.

```

Dim h As Long, m As Long, s As Long, f As Long, tc As Long, str As String
tc = vcx.GetTimecode
str = String(8 - Len(Hex(tc)), "0") + Hex(tc)
s = Val(Mid(str, 5, 2))
m = Val(Mid(str, 3, 2))
h = Val(Mid(str, 1, 2))
f = Val(Mid(str, 7, 2))

```

```
Label1.Caption = h & ":" & m & ":" & s & ":" & f
```

### GetVideoCaps

Return an array of supported video formats

```
Function GetVideoCaps
```



Call this method after Connected has been set to TRUE, but, before starting video preview.  
If device drivers doesn't support this feature, this method Return empty variable.

Returned array has four columns:

1. Video width
2. Video height
3. Bit per pixel value
4. Color format (see ColorFormat property for list of values)

Dim a, f

a = vcx.GetVideoCaps()

vidsize.Clear

For f = LBound(a) To UBound(a)

vidsize.AddItem a(f, 0) & "x" & a(f, 1) & "x" & a(f, 2) & "," & a(f, 3)

Next f

### [GetVideoCodecCount](#)

Return installed video codec count

```
Function GetVideoCodecCount As Long
```

### [GetVideoCodecName](#)

Return video codec name.

```
Function GetVideoCodecName (nIndex As Long) As String
```

### [GetVideoDeviceCount](#)

Return number of video-capture devices in the system

```
Function GetVideoDeviceCount As Long
```

### [GetVideoDeviceDesc](#)

Return video device description

```
Function GetVideoDeviceDesc (Index As Long) As String
```

### [GetVideoDeviceName](#)

Return video device name

```
Function GetVideoDeviceName (Index As Long) As String
```

Index parameter is in range 0 to GetVideoInputCount-1

### GetVideoFormat

Return video size.

```
Function GetVideoFormat (width As Long, height As Long) As Long
```

Return TRUE if successful or FALSE if control is not connected to a capture driver.

ImageHeight and ImageWidth are in pixels.

Dim w As Long, h As Long

DNVideoX.GetVideoFormat w,h

### GetVideoInputCount

Return number of video inputs on currently selected video device (card).

```
Function GetVideoInputCount As Long
```

Return 0 if your video device doesn't have multiple video inputs.

### GetVideoInputName

Return name of specified video channel on multiple-input capture cards.

```
Function GetVideoInputName (Index As Long) As String
```

### GetVideoProcAmpValueRange

Retrieve value range for video property.

```
Function GetVideoProcAmpValueRange (ValueIndex As Long, Min As Long, Max As Long,  
SteppingDelta As Long, Default As Long) As Long
```

ValueIndex parameter is:

Brightness = 0,

Contrast = 1,

Hue = 2,

Saturation = 3,

Sharpness = 4,  
Gamma = 5,  
ColorEnable = 6,  
WhiteBalance = 7,  
BacklightCompensation = 8,  
Gain = 9

Dim Brightness As Long

DNVideoX.GetVideoProcAmp 0, Brightness

Dim MinVal As Long ,MaxVal As Long ,StepD As Long ,DefVal As Long  
DNVideoX.GetVideoProcAmpValueRange 0, MinVal, MaxVal, StepD, DefVal

'set brightness here

Brightness=MinVal+((MaxVal-MinVal)/2)

DNVideoX.SetVideoProcAmp 0, Brightness

### GetVMR9IUnknown

Return VMR9 IUnknown interface if VMR9 is in use. See 'UseVMR9' property.

```
Function GetVMR9IUnknown As Unknown
```

### GrabFrame

Return current video frame as VB Picture object

```
Function GrabFrame As stdole.Picture
```

With this method you can load video frame into PictureBox control.

PictureBox.Picture = DNVideoX.GrabFrame

### HTTPUpload

Use HTTP upload protocol to send information and files to web server

```
Function HTTPUpload (WebServer As String, WebPage As String, Fields As String,  
Files As String)
```

Arguments:

WebServer = web server address

WebPage = name of upload web page

Fields = list of 'fieldname' and 'fieldvalue' values delimited with '|'

Files = list of 'fieldname' and 'file path' values delimited with '|'

Return TRUE if successful, or FALSE otherwise.

```
vcx.HTTPUpload "www.mysite.com" , "upload.asp",  
"field1|value1|field2|value2" ,  
"file1|c:\folder\mypic.jpg|file2|c:\folder\myvideo.avi"
```

### KnobMotorRotate

Control the knob motor's (KM-01) rotation.

```
Function KnobMotorRotate (RotateSpeed As String) As Long
```

RotateSpeed Values:

- 3 : fastest reverse
- 2 : fast reverse
- 1 : slow reverse
- 0 : stop
- 1 : slow forward
- 2 : fast forward
- 3 : fastest forward

Return -1 if knob motor is not connected.

### LoadProfileFromURL

Load WM profile data from .prx file. URL argument must start with 'http://' or 'file://'. Return value is 1 on success or 0 on failure.

```
Function LoadProfileFromURL (URL As String) As Long
```

### PauseCapture

Pause capture

```
Function PauseCapture As Long
```

### PlayRemoteAudio

Connect to remote DNVideoX server and receives only audio stream.

```
Function PlayRemoteAudio (RemoteAddress As String) As Long
```

## ReceiveAudio

Receive audio data packet from server

```
Function ReceiveAudio (ServerAddress As String, Play As Boolean, nChannels As Long,  
nSamplesPerSecond As Long, nBytesPerSample As Long, PCMDData) As Long
```

If you are not interested in PCM data, set last four parameters to 0.

To play a received sound, set Play parameter to TRUE.

Address parameter is a network address of remote DNVideoX server.

```
DNVideoX.ReceiveAudio "127.0.0.1", True, 0, 0, 0, 0
```

## ReceiveFrame

Return video frame from remote server as Picture object.

```
Function ReceiveFrame (ServerName As String) As stdole.Picture
```

Remote server must run DNVideoX control with ServerMode set to TRUE.

Also, ServerPort and ServerPAssword on both computers must be set to identical value.

```
Picture1.Picture = DNVideoX.ReceiveFrame("myvideoserver")
```

## Recompress

Copies AVI/WMV into new file using specified video compression.

```
Function Recompress (SrcFile As String, DestFile As String) As Long
```

If DestFile has an .AVI extension, VideoCodecIndex/AudioCodecIndex settings are used, or, if DestFile has .WMV extension, ProfileIndex/ProfileData settings are used for compression.

## RecompressEx

Use this method to merge video and audio files and/or crop video files.

```
Function RecompressEx (SrcFile1 As String, SrcFile2 As String, DestFile As String,  
TimeStart As Double, TimeEnd As Double) As Long
```

SrcFile1 is source video.

SrcFile2 is source audio MP3, WAV, WMA or AVI file.

DestFile is file to be created.

TimeStart and TimeEnd are time boundaries in milliseconds. If cropping is not needed, set these arguments to 0.

Destination AVI (or WMV) file is compressed using currently selected video/audio codec (or WM profile) .

Cut video file:

```
RecompressEx "original.avi", "", "new.avi", 5000, 15000
```

Merge video and audio into new file.

```
RecompressEx "video.avi", "audio.wav", "new.avi", 0, 0
```

### ResumeCapture

Resume capture paused with PauseCapture method

```
Function ResumeCapture As Long
```

### SaveEDR

Capture picture in EDR (Extended Dynamic Range) mode and save into file.

\* This method is only available for AM4815, 7815, and 7915 series.

```
Function SaveEDR (DeviceIndex As Long, filename As String) As Boolean
```

### SaveEDOF

Capture picture in EDOF (Extended Depth of Field) mode and save into file.

\* This method is only available for AM4815, 7815, and 7915 series.

```
Function SaveEDOF (DeviceIndex As Long, filename As String) As Boolean
```

### SaveFrame

Save video image into file.

```
Function SaveFrame (filename As String) As Boolean
```

### SaveFrameJPG

Save current video frame into JPG file

```
Function SaveFrameJPG (filename As String, quality As Long, Optional resize As Double = 1.0) As Boolean
```

Quality parameter is a JPEG image quality setting (0-100).

Resize is to increase the picture's resolution by interpolating or to decrease it by merging pixels of current video frame. The resize is the ratio of the resolution of saved picture to that of video frame which can only be 2, 1, 0.5, or 0.25.

```
Vcx1.SaveFrameJPG("C:\mypic.jpg", 80) or Vcx1.SaveFrameJPG("C:\mypic.jpg", 80, 2)
```

## SavePictureJPG

Save Picture object into JPG file.

```
Function SavePictureJPG (Picture As stdole.Picture, filename As String, quality As Long) As Long
```

Quality parameter is a JPEG image quality setting (0-100).

If GetHBitmap() method of .NET Bitmap class is used to get bitmap handle, it must be deleted after use to avoid memory leaks.

Example .NET code:

```
[System.Runtime.InteropServices.DllImport("gdi32.dll")]
public static extern bool DeleteObject(IntPtr hObject);

private void axDNVideoX1_NewFrame(object sender, EventArgs e)
{
    pictureBox1.Image = axDNVideoX1.GrabFrame();
    {
        IntPtr hb = ((System.Drawing.Bitmap)(pictureBox1.Image)).GetHbitmap();
        axDNVideoX2.SingleFrameAddPicture(hb.ToInt32());
        DeleteObject(hb);
    }
}
```

```
Vcx1.SaveFrameJPG(Picture1.Picture,"C:\mypic.jpg",80,1)
```

```
Vcx1.SaveFrameJPG(LoadPicture("c:\windows\setup.bmp"),"C:\setup.jpg",90)
```

## SendScriptCommand

Send script type/command pair to broadcast client. This method works only if WM broadcast started with StartBroadcast method is running.

```
Function SendScriptCommand (Type As String, Data As String) As Long
```

Before you call StartBroadcast, you must load a custom WM profile with script stream enabled. See [ProfileData](#) property. To create a custom WM profile, you can use Windows Media Profile Editor tool available for free at Microsoft's web site. If currently selected WM profile doesn't have script stream configured, this method Return -1. On success, this method Return 0.

The following table lists script types that are supported by Windows Media Player.

Script type	Description
URL	The player sends the specified URL to the browser for display to the user. If an embedded player control is being used, you can add a specific frame reference to the URL by using the <code>&amp;&amp;framename</code> syntax.
FILENAME	A URL to another media file to be played.
CAPTION	A text string that is displayed in the captions area of Windows Media Player. This type supports standard HTML formatting, so the text can be formatted as you wish. An example of use is closed captioning.
EVENT	The name of an event that is to occur. The EVENT type supports customization for your own uses. The code for the specified event must be defined in the <a href="#">Windows Media metafile</a> for the stream in order for the player to perform the specified event. An example of use is ad insertion.
OPENEVENT	This script precedes the actual EVENT. The OPENEVENT allows the player to pre-buffer the content so that when the EVENT occurs, the switch between streams appears to be seamless.
TEXT	A TEXT string that is displayed in the captions area of Windows Media Player. Can be plain text, SAMI, or HTML formatted text.

```
vcx.SendScriptCommand "TEXT", "this is a caption / subtitle"
```

### SetAudioDelay

Set audio delay (positive or negative) in captured AVI file. DelayMS argument is in milliseconds.

```
Function SetAudioDelay (DelayMS As Long) As Long
```

### SetAudioFormat

Set audio format for capture

```
Function SetAudioFormat (FmtTag As Long, nChannels As Long, nSamplesPerSec As Long, nAvgBytesPerSec As Long, nBlockAlign As Long, nBitsPerSample As Long) As Boolean
```



## FormatTag

Waveform-audio format type. Format tags are registered with Microsoft Corporation for many compression algorithms. A complete list of format tags can be found in the MMREG.H header file.  
WAVE\_FORMAT\_PCM = 1

## nChannels

Number of channels in the waveform-audio data. Monaural data uses one channel and stereo data uses two channels.

## nSamplesPerSec

Sample rate, in samples per second (hertz), that each channel should be played or recorded. If wFormatTag is WAVE\_FORMAT\_PCM, then common values for nSamplesPerSec are 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz. For non-PCM formats, this member must be computed according to the manufacturer's specification of the format tag.

## nAvgBytesPerSec

Required average data-transfer rate, in bytes per second, for the format tag. If wFormatTag is WAVE\_FORMAT\_PCM, nAvgBytesPerSec should be equal to the product of nSamplesPerSec and nBlockAlign. For non-PCM formats, this member must be computed according to the manufacturer's specification of the format tag.

Playback and record software can estimate buffer sizes by using the nAvgBytesPerSec member.

## nBlockAlign

Block alignment, in bytes. The block alignment is the minimum atomic unit of data for the wFormatTag format type. If wFormatTag is WAVE\_FORMAT\_PCM, nBlockAlign should be equal to the product of nChannels and wBitsPerSample divided by 8 (bits per byte). For non-PCM formats, this member must be computed according to the manufacturer's specification of the format tag.

Playback and record software must process a multiple of nBlockAlign bytes of data at a time. Data written and read from a device must always start at the beginning of a block. For example, it is illegal to start playback of PCM data in the middle of a sample (that is, on a non-block-aligned boundary).

## nBitsPerSample

Bits per sample for the wFormatTag format type. If wFormatTag is WAVE\_FORMAT\_PCM, then wBitsPerSample should be equal to 8 or 16. For non-PCM formats, this member must be set according to the manufacturer's specification of the format tag. Note that some compression

schemes cannot define a value for `wBitsPerSample`, so this member can be zero.

### SetAudioInputLevel

Set the recording level for audio input selected with `AudioInputIndex` property. Level value is in range 0 to 100.

```
Function SetAudioInputLevel (Level As Long) As Long
```

Call this method after you have set `Connected` property to `TRUE` and after you set `AudioInputIndex` property.

### SetAudioVolume

Set audio renderer volume. Volume argument range is 0 - 100. if Volume is -1, this method Return current audio volume.

```
Function SetAudioVolume (Volume As Long) As Long
```

### SetAutoExposure

Turn `AutoExposure` on or off of current video frame.

```
Function SetAutoExposure (DeviceIndex As Long, AutoExposure As Long)
```

`AutoExposure` value:

0=AE off

1=AE on

### SetAWBB

Set AWB Blue setting of current video frame.

\* This method is only available for Premier and Edge series.

```
Function SetAWBB (DeviceIndex As Long, AWBB as Long)
```

AWBB parameter is in range 0 to 255.

Note: For 5M Premier and 5M Edge series, it is needed to enable the `FreezeAWB` before using `SetAWBB`.

### SetAWBG

Set AWB Green setting of current video frame.

\* This method is only available for Premier and Edge series excepting AM4011 and 4013 models.

```
Function SetAWBG (DeviceIndex As Long, AWBG As Long)
```

AWBG parameter is in range 0 to 255.

Note: For 5M Premier and 5M Edge series, it is needed to enable the FreezeAWB before using SetAWBG.

### SetAWBR

Set AWB Red setting of current video frame.

\* This method is only available for Premier and Edge series.

```
Function SetAWBR (DeviceIndex As Long, AWBR As Long)
```

AWBR parameter is in range 0 to 255.

Note: For 5M Premier and 5M Edge series, it is needed to enable the FreezeAWB before using SetAWBR.

### SetBitmapOverlay

Set bitmap to show on-video

```
Function SetBitmapOverlay (BitmapHandle As Long, x As Long, y As Long, TransColor  
As Long, Alpha As Long) As Long
```

BitmapHandle parameter is a Windows handle of the bitmap.

For standard PictureBox control, use Picture.Handle property to get this value.

TransColor parameter is a RGB value of transparent color, if no transparency is used, set this parameter to -1.

Alpha paramter is in range 0 (transparent) to 255 (opaque) .

Use SetBitmapOverlay 0,0,0,0,0 to remove bitmap overlay.

DNVideoX.SetBitmapOverlay Picture1.Picture.Handle, 0, 0, -1,127

### SetChromaKey

Set chrom-key effect. BackImage parameter is a filename of static image background which will be visible through video. Color paramters are RGB values.

```
Function SetChromaKey (BackImage As String, MinTransparentColor As Long,  
MaxTransparentColor As Long) As Long
```

### SetCrop

Crop live-video stream to the rectangle of dimensions (W,H) and with top-left coordinate of (X,Y).

```
Function SetCrop (x As Long, y As Long, W As Long, H As Long) As Long
```

This method must be called before connecting DNVideoX to video source.

To capture a face in front of camera (if camera has 320x240 resolution), use:  
DNVideoX.SetCrop 110, 45, 100, 150

### SetAETarget

Set Auto Exposure Target value of current video frame.

```
Function SetAETarget (DeviceIndex As Long, Exposure As Long)
```

The AETarget can only be set when the AutoExposure is set to ON.  
The Value can be set from 16 to 220.

### SetAESTability

Set Auto Exposure's Stability value of current video frame.

\* This method is only available for AM/AD-3011, 3013, 4011, 4013, 4113, and 4023 series.

```
Function SetAESTability (DeviceIndex As Long, StabilityValue) As Long
```

The ExposureStability value can be set from 1 to 16. The higher the ExposureStability value is, the less accurate the auto exposure control will be.

### SetExposureValue

Set relative exposure value, which is proportional to the exposure time, of current video frame.

\* This method is only available for Premier and Edge series.

```
Function SetExposureValue (DeviceIndex As Long) As Long
```

The ExposureValue can only be set when the AutoExposure is set to OFF.

The range of ExposureValue is varied with different series:

<u>Series</u>	<u>Range of Exposure Value</u>
3011, 3013	8 to 30612
1.3M Premier	1 to 41771
1.3M Edge	1 to 63076
5M Premier / 5M Edge	1 to 30000

### SetExposureTime

Set Exposure Time manually. The property value is expressed in log base 2 seconds, thus, for values less than zero, the exposure time is  $1/2^n$  seconds. For positive values and zero, the exposure is  $2^n$  seconds.

\* This method is only available for Premier and Edge series.

```
Function SetExposureTime (DeviceIndex As Long, Value As Long) As Long
```

The ExposureValue can only be set when the AutoExposure is set to OFF.

Value	Seconds
-7	1/128
-6	1/64
-5	1/32
-4	1/16
-3	1/8
-2	1/4
-1	1/2
0	1
1	2

### SetFLCSwitch

Set the LED quadrant switch to partially turn on the LEDs.

\* This method is only available for FLC equipped Dino-Lite, such as AM7115 and AM7515.

```
Function SetFLCSwitch (DeviceIndex As Long, Value As Long)
```

FLC switch argument values:

<u>Value</u>	<u>Switch-on Quadrant</u>
0	All LEDs turn off

1	1
2	2
3	1, 2
4	3
5	1, 3
6	2, 3
7	1, 2, 3
8	4
9	1, 4
10	2, 4
11	1, 2, 4
12	3, 4
13	1, 3, 4
14	2, 3, 4
15	1, 2, 3, 4

### SetFLCLevel

Set the LED brightness level (range 0 to 6).

\* This method is only available for FLC equipped Dino-Lite, such as AM7115 and AM7515.

**Function SetFLCLevel (DeviceIndex As Long, Value As Long)**

### SetGain

Set Gain of current video frame.

\* This method is only available for Premier and Edge series.

**Function SetGain (DeviceIndex As Long) As Long**

The Gain can only be set when the AutoExposure is set to OFF.

The range of Gain is varied with different series:

<u>Series</u>	<u>Range of Gain</u>
3011, 3013	0 to 255
1.3M Premier	0 to 362
1.3M Edge	0 to 382
5M Premier / 5M Edge	0 to 47

### SetFadeLevel

Set fade level (the range is 0-100, 0 is neutral) for video

**Function SetFadeLevel (NewLevel As Long) As Long**

## SetFilterSettings

Set filter settings

```
Function SetFilterSettings (Filter As Long, Data As String) As Long
```

Filter argument values:

- 1 = video compress filter
- 2 = audio compress filter
- 3 = video source filter
- 4 = audio source filter
- 5 = user filter

This method Return 0 on success or negative value on error.

Data parameter is string retrieved by GetFiltlerSettings method. Make sure that same filter is selected as when GetFilterSettings is called. Every fityler Return/accepts different format of settings binary data.

'before this, load filter data in string variable d

DNVideoX.VideoCodecIndex = 6

DNVideoX.SetFitlerSettings 1, d

## SetFlicker

Set flicker reduction frequency of current video frame

\* This method may not be applicable to AM/AD 211, 3613, and 3713 series.

```
Function SetFlicker (Flicker As Long As Long)
```

Flicker values:

- 1 : 60Hz
- 2 : 50Hz

## SetHighPriority

Set priority class for the current process. High argument is TRUE for high priority or FALSE for normal priority.

```
Function SetHighPriority (High As Boolean) As Boolean
```

## SetLEDState

Switch the camera LED.

\* The LEDState will be controllable only when the camera preview is established.

\* This method may not be applicable to AM211, AM2011, and Dino-Eye series.

```
Function SetLEDState(DeviceIndex As Long, LEDState As Long) As Long
```

LEDState Values:

0 = LED off

1 = LED1 on

2 = LED2 on. The LED2 only exists on models with 2 switchable LEDs.

### SetLEDStrobeLength

Set the LED strobe length of 3713TB series models from 1 to 16. This method is only available for AM/AD-3713TB.

```
Function SetLEDStrobeLength (DeviceIndex As Long, StrobeLength As Long) As Long
```

The StrobeLength can be set from 1 to 16.

### SetMasterAudioVolume

Set master volume for specified mixer line. Volume argument range is 0 - 100. if Volume is -1, this method Return current audio volume.

```
Function SetMasterAudioVolume (LineID As Long, Volume As Long) As Long
```

LineID values:

DST_DIGITAL	1
DST_LINE	2
DST_MONITOR	3
DST_SPEAKERS	4
DST_HEADPHONES	5
DST_TELEPHONE	6
DST_WAVEIN	7
DST_VOICEIN	8
SRC_DIGITAL	11
SRC_LINE	12
SRC_MICROPHONE	13
SRC_SYNTHESIZER	14
SRC_COMPACTDISC	15



SRC_TELEPHONE	16
SRC_PCSPEAKER	17
SRC_WAVEOUT	18
SRC_AUXILIARY	19
SRC_ANALOG	20

### SetMirror

Set Mirror to the current video frame.

\* This method is only available for AM/AD-3011, 3013, 4011, 4013, 4113, 4023, 4115, 4515 and 4815 series.

```
Function SetMirror (DeviceIndex As Long, Mirror As Long)
```

Mirror value:

0=normal

1=vertical mirror

2=horizontal mirror

3=vertical +horizontal mirror

### SetMotionMask

Set rectangle(s) to ignore on the image while detecting motion.

```
Function SetMotionMask (Index As Long, left As Long, top As Long, width As Long,  
height As Long) As Long
```

Index parameter is in range 0 - 9. Dimmensions are in pixels.

### SetTextOverlay

Set on-video text caption.

```
Function SetTextOverlay (Index As Long, Caption As String, x As Long, y As Long,  
FontName As String, FontSize As Long, FColor As Long, BColor As Long) As Long
```

Return TRUE on succes, FALSE otherwise.

ID parameter is in range 0 to 19.

X, Y and FontSize parameters are in device pixels.

Use VisualBasic RGB function to set TextColor and TextBgColor parameters.

Set TextBGColor parameter to -1 for transparent text output.

**Special values are "TIME" for Text parameter to show time-stamp and "SMPTE" for SMPTE-format time display.**

To clear text caption, use SetTextOverlay method with an empty string as Text parameter.

To set date-time stamp in top-left corner on video, use:

```
DNVideoX.SetTextOverlay 0, "TIME", 0, 0, "Arial", 14, RGB(255,0,0), -1
```

### SetVideoFormat

Set video image dimensions

```
Function SetVideoFormat (width As Long, height As Long) As Boolean
```

Return TRUE if successful or FALSE otherwise.

Because video formats are device-specific, applications should check the return value from this function to determine if the format is accepted by the driver.

### SetVideoFormatEx

Set video format by capability index returned by GetVideoCaps method

```
Function SetVideoFormatEx (CapabilityIndex As Long) As Long
```

### SetZoom

Set zoom rectangle on video. Use all zeros as parameters to this method to reset zoom.

```
Function SetZoom (left As Long, top As Long, width As Long, height As Long) As Long
```

Dim w, h

w = vcx.VideoWidth / 2

h = vcx.VideoHeight / 2

vcx.SetZoom w / 2, h / 2, w, h

### ShowAudioCodecDlg

Shows audio codec dialog.

`Function ShowAudioCodecDlg As Long`

Display a settings dialog provided by video/audio device driver. Must be connected to use this method.

### ShowAudioFormatDlg

Shows audio format dialog.

`Function ShowAudioFormatDlg As Long`

Display a settings dialog provided by video/audio device driver. Must be connected to use this method.

### ShowAudioSourceDlg

Show audio source dialog.

`Function ShowAudioSourceDlg As Long`

Display a settings dialog provided by video/audio device driver. Must be connected to use this method.

### ShowBarcodeSettingDlg

Display a barcode reading setting dialog.

`Function ShowBarcodeSettingDlg As Long`

### ShowUserFilterDlg

Display user filter property page

`Function ShowUserFilterDlg (FilterIndex As Long) As Long`

### ShowVideoCodecDlg

Show video codec dialog.

`Function ShowVideoCodecDlg As Long`

Display a settings dialog provided by video/audio device driver. Must be connected to use this method.

### ShowVideoCrossbarDlg

Display video crossbar dialog

```
Function ShowVideoCrossbarDlg As Long
```

### ShowVideoFormatDlg

Show video format dialog.

```
Function ShowVideoFormatDlg As Long
```

Display a settings dialog provided by video/audio device driver. Must be connected to use this method.

### ShowVideoSourceDlg

Show video source dialog.

```
Function ShowVideoSourceDlg As Long
```

Display a settings dialog provided by video/audio device driver. Must be connected to use this method.

### SingleFrameAdd

Adds current video frame into AVI file opened by SingleFrameOpen method

```
Function SingleFrameAdd As Long
```

Preview must be enabled.

Single frame capture and real-time capture can't run in the same time.

### SingleFrameAddPicture

Adds a Windows bitmap to the AVI file created with SingleFrameAdd method.

```
Function SingleFrameAddPicture (BitmapHandle As Long) As Long
```

If GetHBitmap() method of .NET Bitmap class is used to get bitmap handle, it must be deleted after use to avoid memory leaks.

Example:

```
[System.Runtime.InteropServices.DllImport("gdi32.dll")]
public static extern bool DeleteObject(IntPtr hObject);

private void axDNVideoX1_NewFrame(object sender, EventArgs e)
{
    pictureBox1.Image = axDNVideoX1.GrabFrame();
    {
        IntPtr hb = ((System.Drawing.Bitmap)(pictureBox1.Image)).GetHbitmap();
        axDNVideoX2.SingleFrameAddPicture(hb.ToInt32());
        DeleteObject(hb);
    }
}
```

### SingleFrameClose

Closes single-frame AVI capture

```
Function SingleFrameClose As Long
```

### SingleFrameOpen

Creates AVI file for single-frame capturing.

```
Function SingleFrameOpen (fps As Long) As Long
```

Preview must be enabled.

Single frame capture and real-time capture can't run in the same time.

AVI filename is specified by CapFilename property of DNVideoX control.

Fps parameter Set frames-per-second value in new AVI file. Use SingleFrameOpen(5) for 5 frames-per-second AVI.

### StartBroadcast

Starts WindowsMedia network broadcast at specified port. Use Windows MediaPlayer's OpenURL command to see video on network.

```
Function StartBroadcast (port As Long, MaxConnections As Long) As Long
```

This method starts Windows Media broadcast from local PC.

Parameter:

port specifies TCP/IP port number to use for broadcast.

MaxConnections specifies how many clients can connect.

Before starting broadcast, you should select WM profile (bitrate, etc.) by setting ProfileIndex (ProfileData) property.

When a client (Windows Media Player) connects, ConnectionRequest event is raised. When clients ends connection, ConnectionClosed event is raised.

```
vcx.Connected=TRUE
```

```
vcx.Preview=TRUE
```

```
vcx.WMAttributes="my title|author is me|copyright to me|rating is 5|this is description "
```

```
vcx.StartBroadcast 8080,5
```

### StartBroadcastPush

Start sending broadcast to Windows Media server publishing point

```
Function StartBroadcastPush (URL As String, User As String, password As String) As Long
```

String that contains the URL of the publishing point on the Windows Media server. For example, if the URL is "http://MyServer/MyPublishingPoint", the push sink connects to the publishing point named MyPublishingPoint on the server named MyServer. The default port number is 80. If the server is using a different port, specify the port number in the URL. For example, "http://MyServer:8080/MyPublishingPoint" specifies port number 8080.

If the publishing point specified in pwsURL does not exist, the server creates a new publishing point. The caller must have write and create permissions on the server. The new publishing point has the same configuration as the server's default publishing point.

Use Username and Password parameters to authorize to Windows Media Server.

```
vcx.WMAttributes="my title|author is me|copyright to me|rating is 5|this is description "
```

```
vcx.StartBroadcastPush "http://myserver:8080/pubpoint","mylogin","mypassword"
```

### StartCapture

Starts video capture

```
Function StartCapture As Boolean
```

Return TRUE if successful or FALSE otherwise.

Captured data is saved into file specified in CapFilename property.

Video can be captured in AVI or WMV files. If CapFilename property has .AVI extension, audio/video codec can be specified using AudioCodecIndex/VideoCodecIndex properties.

If .WMV file is being captured, audio/video compression is determined by ProfileIndex or ProfileData properties.

WM stream attributes can be set using WMAttributes property.

### StopBroadcast

Stops WM broadcast

```
Function StopBroadcast As Long
```

### StopCapture

Stops video capture

```
Function StopCapture As Boolean
```

### UploadFile

Upload a file to FTP server

```
Function UploadFile (server As String, username As String, password As String, path  
As String, server_filename As String, local_filepath As String, [port As Long = 21])  
As Boolean
```

### UploadFrame

Sends current video frame to FTP server

```
Function UploadFrame (server As String, username As String, password As String,  
path As String, filename As String, port As Long, quality As Long) As Boolean
```

Quality parameter is a JPEG image quality setting (0-100).

Port is usually 21 for FTP service.

See FTPPassiveMode property.

```
Vcx1.UploadFrame("ftp.foo.com","john","tiger","images","mypic.jpg",21,70)
```

### CaptureEnd

Triggered when capture is ended

```
Sub CaptureEnd
```

### CaptureReady

Raised after StartCapture is called but before any video is actually captured into file. It gives application the opportunity to display 'press to start capture...' message.

`Sub CaptureReady`

### CaptureStart

Triggered when capture is started

`Sub CaptureStart`

### DeviceLost

Raised when device lost is detected. Such as camera removal or cable plug-out.

`Sub DeviceLost`

### FootPedalPressed

This event is fired when button on the foot pedal is pressed.

`Sub FootPedalPressed`

### FullscreenLost

Raised when full-screen mode ends due to user action

`Sub FullscreenLost`

### NewFrame

Raised when new video frame is available

`Sub NewFrame`

### RecompressCompleted

Recompress method runs in background. This event is fired when recompression is finished.

`Sub RecompressCompleted`



### RecompressProgress

Reports progress of Recompress method processing.

```
Sub RecompressProgress (nPercent As Long, Cancel As Long)
```

### MicroTouchPressed

This event is fired when MicroTouch button on the camera is pressed.

```
Sub MicroTouchPressed
```

### Pix2Length

Converts pixel to length for camera

```
Function Pix2Length (fPixel As Double, fMag As Double, vWidth As Long, DeviceIndex  
As Long) As Double
```

This method converts the number of pixels on the screen of preview window to the dimensional length (unit: um).

fPixel argument is the number of pixels on the video preview window to be converted

fMag argument is the magnification value

vWidth argument is the width of the video preview window

DeviceIndex argument is the camera index

### Pix2Length2

Converts pixel to length for picture

```
Function Pix2Length2 (fPixel As Double, fMag As Double, pWidth As Long, ProductName  
As String) As Double
```

This method converts the number of pixels to dimensional length (unit: um) for picture taken with known Dino-Lite or Dino-Eye product.

fPixel argument is the number of pixels to be converted

fMag argument is the magnification value

pWidth is the resolution width of the picture, e.g. 640 is the width of picture with 640x480 resolution.

ProductName is the short name of the supported Dino-Lite or Dino-Eye in the following list.

313, 413, 3003, 3013, 3613, 3713, 4013, 4113, 4023, 4115, 4515, 4815, 7013, 7023, 7115, 7515, 7815, 7915, AMH, AMH2