



ActiveX Control Specification

Version	Date	Changes
1.3	2010/12/14	Modify for GetCurrentImage ()
1.2	2010/11/30	Modify for Table of Contents
1.1	2010/6/23	Modify for 3.49 Select()
1.0	2010/5/27	Created



Table of Contents

1. Introduction	6
2. Requirements	6
3. Properties	6
3.1. AutoStart	6
3.2. AudioTransEnc	7
3.3. AxC	8
3.4. EnableReconnect	8
3.5. MediaFilePath	9
3.6. MediaFilePrefix	9
3.7. MediaInfoEnabled	9
3.8. MediaPassword	10
3.9. MediaType	10
3.10. MediaURL	11
3.11. MediaUsername	11
3.12. Mute	11
3.13. Rotation	12
3.14. Status	12
3.15. UIMode	13
3.16. CfgMDArea	13
3.17. AutoSaveClientCfg	14
3.18. SnapshotFilePath	14
3.19. MediaFileFormat	15
3.20. JitterBufTime	15
3.21. SnapshotFilePrefix	15
3.22. ForceGDI	16
3.23. ForceMulticast	16
3.24. EnableOverlay	16



3.25. EnablePopupRecord.....	16
3.26. EnablePopupSnapshot.....	17
3.27. FullScreen	17
3.28. PTZControlURL (Not Implement)	17
3.29. MotionConfigURL (Not Implement).....	17
3.30. MotionDataURL (Not Implement)	18
3.31. AudioTransmitType	18
3.32. EnableGetAudio.....	18
3.33. EnableGetVideo	19
3.34. EnableGetEvent	19
3.35. BackgroundBMP.....	20
3.36. ConnectingBMP.....	20
3.37. IsConnected.....	20
3.38. SkipPlay	21
3.39. PreBufferRecordTime	21
3.40. MinRecordingTime	21
3.41. MaxRecordingTime	22
3.42. MDMode.....	22
3.43. EnableUNCAuthDialog	23
3.44. RecordUNCPATHUsername	23
3.45. RecordUNCPATHPassword	24
3.44. SnapshotUNCPATHUsername	24
3.45. SnapshotUNCPATHPassword	24
3.46. AudioCodec	25
3.47. VideoCodec.....	26
3.48. Focus	26
3.49. Select.....	27
3.50. ShowStatusBar	27



3.51. BorderSize.....	27
4. Methods.....	28
4.1. AudioTransmitStart()	28
4.2. AudioTransmitStop()	28
4.3. Play()	29
4.4. RequestService()	29
4.5. SaveCurrentImage()	30
4.6. SetRecLimit().....	31
4.7. StartRecord()	31
4.8. Stop().....	32
4.9. StopRecord()	32
4.10. ZoomDigitally()	32
4.11. GetPort().....	33
4.12. SetPort()	33
4.13. GetMirror().....	34
4.14. SetMirror()	35
4.15. ShowFileFolderDialog()	36
4.16. SetMDArea().....	36
4.17. GetMDAreaX()	37
4.18. GetMDAreaY()	37
4.19. GetMDAreaWidth()	38
4.20. GetMDAreaHeight()	38
4.21. ControlDecoding()	39
4.22. LoadClientCfg()	39
4.23. SaveClientCfg()	40
4.24. ApplyAllSavedClientCfg()	40
4.25. GetRGB()	41
4.26. GetVideoRenderInfo()	41



4.27. MouseWheel()	42
4.28. SetZoomInLimit()	42
4.29. GetZoomInLimit()	43
4.30. SetBackground()	43
4.31. SetFullScreenEnabled()	44
4.32. SetRenderFPSFactor()	45
4.33. GetCurrentImage()	45
4.34. SetAudioTransmitData()	46
4.35. SetDisplayText()	46
4.36. SaveCurrentImageEx()	47
4.37. StartRecordEx()	47
4.38. StopRecordEx()	48
5. Events	49
5.1. OnClick()	49
5.2. OnError()	49
5.3. OnMediaInfo()	50
5.4. OnMouseDown()	51
5.5. OnMouseMove()	52
5.6. OnMouseUp()	52
5.7. OnRequestResponded()	53
5.8. OnStatusChange()	53
5.9. OnDoubleClick()	54
5.10. OnMouseWheel()	54
5.11. OnEvent()	55
5.12. OnReconnect()	56
5.13. OnMediaRecorded()	57
5.14. OnFullScreen()	57
5.15. OnRecord()	57



5.16. OnSnapshot()	58
5.17. OnNewAudioFrame()	58
5.18. OnNewVideoFrame().....	59
5.19. OnNewEventFrame()	59
5.20. OnNewEventFrameEx().....	60
5.21. OnEventEx()	63
5.22. OnRTPAudioFrame().....	63
5.23. OnRTPVideoFrame().....	64
5.23. OnRTPEventFrame()	65

1. Introduction

The ActiveX control is designed for 3S video servers and IP cameras. It provides easy interfaces to integrate these networking multi-media devices into your applications.

2. Requirements

Hardware: Intel P4 1GHz CPU with 256MB RAM or above.

OS: Windows 2000/SP4 with IE 5.0 or above.

3. Properties

3.1. AutoStart

Set the property TRUE, if the ActiveX control should connect to the server automatically after media URL is given.

Definition

BOOL AutoStart [FALSE]

Remarks

None.



3.2. AudioTransEnc

Specify the audio codec format to transmit.

Definition

LONG AudioTransEnc [2]

Value	Description
2	G.711a
3	G.711u
4	G.726 32Kbps
5	G.726 24Kbps

Remarks

None.

3.3. AxC

Enable/disable ANC/AEC/AGC functions.

Definition

LONG AxC [0]

Value	Description
0	None.
1	ANC.
2	AEC. Not implemented.
4	AGC. Not implemented.

Remarks

None.

3.4. EnableReconnect

Set the property to be TRUE, if the ActiveX control should reconnect when the streaming connection is lost.

Definition

BSTR MediaFilePath [""]

Remarks

None.



3.5. MediaFilePath

Set the directory path to store media files.

Definition

BSTR MediaFilePath [""]

Remarks

None.

3.6. MediaFilePrefix

Set the prefix of a media filename.

Definition

BSTR MediaFilePrefix [""]

Remarks

None.

3.7. MediaInfoEnabled

Indicate if the MediaInfo event should be fired every second.

Definition

BOOL MediaInfoEnabled [FALSE]

Remarks

None.



3.8. MediaPassword

Set the password for the streaming access.

Definition

BSTR MediaPassword [""]

Remarks

None.

3.9. MediaType

Set the streaming type.

Definition

BSTR MediaType [「udp」]

Value	Description
udp	RTP/RTCP using UDP.
rtsp or tcp	RTP/RTCP over RTSP/TCP tunnel.
http	RTP/RTCP/RTSP over HTTP tunnel.

Remarks

None.



3.10. MediaURL

Set the streaming URL.

Definition

BSTR MediaURL [""]

Remarks

The ActiveX control connects the URL to receive streaming media.

3.11. MediaUsername

Set the username for the streaming access.

Definition

BSTR MediaUsername [""]

Remarks

None.

3.12. Mute

Set to turn on/off audio.

Definition

BOOL Mute [FALSE]

Remarks

None.



3.13. Rotation

Rotate the image in degree.

Definition

LONG Rotation [0]

Remarks

Now only 0 and 180 is valid value.

3.14. Status

Read only bit mask. The property indicates the status of the control.

Definition

LONG Status [1]

Value	Description
1	Idle.
2	Playing.
8	Recording.

Remarks

None.

3.15. UI Mode

Set the UI mode of the control.

Definition

BSTR Mute [「 none 」]

Value	Description
none	Normal UI mode.
MDCConfig	The user can config the motion detection windows.

Remarks

None.

3.16. CfgMDArea

Enable or disable motion detection area.

Definition

LONG CfgMDArea[0]

Value	Description
0	Disable all motion detection areas.
> 0	Enable motion detection area.

Remarks

The max number of configurable areas is 4.



3.17. AutoSaveClientCfg

Auto save client configurations.

Definition

BOOL AutoSaveClientCfg[FALSE]

Value	Description
TRUE	Auto save.
FALSE	None auto save.

Remarks

If set TRUE, MediaType, MediaFilePath, MediaFilePrefix, SnapshotFilePath, SnapshotFilePrefix will be saved locally.

One can retrieve the previous configurations next time.

3.18. SnapshotFilePath

Set snapshot file path.

Definition

BSTR AutoSaveClientCfg[".\snapshot"]

Remarks

None.



3.19. MediaFileFormat

Set media file format for recording.

Definition

LONG MediaFileFormat[2]

Value	Description
0	error type
1	MP4
2	AVI

Remarks

MP4 is preferred in the future.

3.20. JitterBufTime

Get/Set jitter buffer time for A/V synchronization.

Definition

LONG JitterBufTime[150]

Remarks

Value ranges from 0, 50ms, 150ms -10 sec.

3.21. SnapshotFilePrefix

Set snapshot file prefix.

Definition

BSTR SnapshotFilePrefix[\""]

Remarks

None.



3.22. ForceGDI

Force using Windows GDI rendering mode.

Definition

BOOL ForceGDI[FALSE]

Remarks

Enable this will create RGB renderer.

3.23. ForceMulticast

Force set to multicast.

Definition

BOOL ForceMulticast[FALSE]

Remarks

MediaType should be set "udp".

3.24. EnableOverlay

Enable YUV overlay rendering mode by default.

Definition

BOOL EnableOverlay[FALSE]

Remarks

None.

3.25. EnablePopupRecord

Enable Popup menu record option .

Definition

BOOL EnablePopupRecord[FALSE]

Remarks

file name depend on "MediaFilePath" "MediaFilePrefix"



3.26. EnablePopupSnapshot

Enable Popup menu snapshot option .

Definition

BOOL EnablePopupSnapshot[FALSE]

Remarks

file name depend on "SnapshotFilePath" "SnapshotFilePrefix"

3.27. FullScreen

Enable or Disable FullScreen.

Definition

BOOL FullScreen[FALSE]

Remarks

Same function as SetFullScreenEnabled method.

3.28. PTZControlURL (Not Implement)

Control PTZ CGI.

Definition

BSTR PTZControlURL[\""]

Remarks

Same function as RequestService method.

3.29. MotionConfigURL (Not Implement)

Set motion detect area.

Definition

BSTR MotionConfigURL[\""]

Remarks

Same function as SetMDArea method.



3.30. MotionDataURL (Not Implement)

None

Definition

BSTR MotionDataURL[""]

Remarks

None.

3.31. AudioTransmitType

Select audio upload type.

Definition

LONG AudioTransmitType [0]

Value	Description
0	From HW Capture
1	From SetAudioTransmitData()

Remarks

None.

3.32. EnableGetAudio

Enable OnNewAudioFrame() Event.

Definition

BOOL EnableGetAudio[FALSE]

Value	Description
FALSE	Disable OnNewAudioFrame()
TRUE	Enable OnNewAudioFrame()

Remarks

None.



3.33. EnableGetVideo

Enable OnNewVideoFrame() Event.

Definition

BOOL EnableGetVideo[FALSE]

Value	Description
FALSE	Disable OnNewVideoFrame()
TRUE	Enable OnNewVideoFrame()

Remarks

None.

3.34. EnableGetEvent

Enable OnNewEventFrame() Event.

Definition

BOOL EnableGetVideo[FALSE]

Value	Description
FALSE	Disable OnNewEventFrame()
TRUE	Enable OnNewEventFrame()

Remarks

None.



3.35. BackgroundBMP

Set background bitmap.

Definition

BSTR BackgroundBMP[""]

Remarks

None.

3.36. ConnectingBMP

Set connecting background bitmap.

Definition

BSTR ConnectingBMP[""]

Remarks

None.

3.37. IsConnected

Is connected.

Definition

BOOL IsConnected[FALSE]

Remarks

None.



3.38. SkipPlay

Connect to server without play Video and Audio

Definition

BOOL SkipPlay[FALSE]

Remarks

None.

3.39. PreBufferRecordTime

Set record pre-buffer time.

Definition

LONG PreBufferRecordTime[3000]

Remarks

Value ranges from 0 -10000ms.

3.40. MinRecordingTime

Set record Min. record time.

Definition

LONG MinRecordingTime[10000]

Remarks

Value ranges from 10s =>MaxRecordingTime.



3.41. MaxRecordingTime

Set record Max. record time.

Definition

LONG MaxRecordingTime[30000]

Remarks

Value ranges from MinRecordingTime. => Unlimit

3.42. MDMode

Set Motion Detect Window Mode

Definition

LONG MDMode[0]

Value	Description
0	default
1	Lock at screen center, edges scale at same time.
2	edges scale at same time.

Remarks

default mode 0.

3.43. EnableUNCAuthDialog

Enabled to auto pop-up UNC authorization dialog if the snapshot or recording use the UNC path that need necessary rights. The values of username or password of properties will be changed after assigning the new values and pressing the OK button in dialog.

Definition

BOOL EnableUNCAuthDialog[FALSE]

Value	Description
False	Disabled
True	Enabled

Remarks

Default is disabled.

3.44. RecordUNCPathUsername

The username of UNC path authorization is used by recording path

Definition

BSTR RecordUNCPathUsername[\"\"]

Remarks

Do not suggest using it, please establish the share folder connection in advance like as mapping local device.



3.45. RecordUNCPathPassword

The password of UNC path authorization is used by recording path

Definition

BSTR RecordUNCPathPassword [""]

Remarks

Do not suggest using it, please establish the share folder connection in advance like as mapping local device.

3.44. SnapshotUNCPathUsername

The username of UNC path authorization is used by snapshot path

Definition

BSTR SnapshotUNCPathUsername [""]

Remarks

Do not suggest using it, please establish the share folder connection in advance like as mapping local device.

3.45. SnapshotUNCPathPassword

The password of UNC path authorization is used by snapshot path

Definition

BSTR SnapshotUNCPathPassword [""]

Remarks

Do not suggest using it, please establish the share folder connection in advance like as mapping local device.



3.46. AudioCodec

Get Audio Codec Type

Definition

long AudioCodec [0]

Remarks

AENC_PCM = 1,

AENC_G711A = 2,

AENC_G711U = 3,

AENC_G726_32 = 4,

AENC_G726_24 = 5,

AENC_G722_64 = 6,

AENC_G722_56 = 7,

AENC_G722_48 = 8,

AENC_AMR = 9,

AENC_AAC = 10,



3.47. VideoCodec

Get Video Codec Type

Definition

long VideoCodec[0]

Remarks

VENC_MPEG4 = 1

VENC_MJPEG = 2

VENC_H263 = 3

VENC_H264 = 4

3.48. Focus

Set Focus status.

Definition

BOOL Focus[false]

Remarks

Only remberer status. Do nothing.



3.49. Select

Set Select status.

Definition

BOOL Select[false]

Remarks

Only remember status. Do nothing.

3.50. ShowStatusBar

Display status bar

Definition

BOOL ShowStatusBar[false]

Remarks

Not implement.

3.51. BorderSize

Set border size

Definition

long BorderSize[0]

Remarks

Not implement.



4. Methods

4.1. *AudioTransmitStart()*

Start to transmit audio streaming to the server.

Definition

BOOL AudioTransmitStart()

Parameters

None.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

Be sure to set AudioTransEnc first.

4.2. *AudioTransmitStop()*

Stop transmitting audio streaming to the server.

Definition

BOOL AudioTransmitStop()

Parameters

None.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.3. Play()

Connects to the remote device and play the media streaming.

Definition

BOOL Play()

Parameters

None.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.4. RequestService()

Request for a service provided by the server.

Definition

LONG RequestService(BSTR *uri*, BSTR *body*)

Parameters

uri[in] The service URI. It can be followed by a query string, e.g. /param.cgi?action=list.

body[in] The default value is null string(e.g. "").

Return Value

A positive number for UID if the method call is successful. Otherwise, 0 returned.

Remarks

Ex.

PTZ control uri (%d is speed var)

UP : /ptz.cgi?move=up&speed=%d

DOWN : /ptz.cgi?move=down&speed=%d

LEFT : /ptz.cgi?move=left&speed=%d

RIGHT : /ptz.cgi?move=right&speed=%d

ZOOM IN : /ptz.cgi?rzoom=4&speed=4

ZOOM OUT : /ptz.cgi?rzoom=-4&speed=4

ZOOM STOP : /ptz.cgi?rzoom=0

PT HOME : /ptz.cgi?move=home&speed=%d

4.5. **SaveCurrentImage()**

Save the current frame into an image file.

Definition

BOOL SaveCurrentImage(int *format*, BSTR *filename*)

Parameters

format[in] The image format.

Value	Description
0	JPEG.

filename[in] The filename of the saved iamge.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.6. SetRecLimit()

Set a type of limit for media batch recording.

Definition

LONG setRecLimit(LONG *type*, LONG *value*)

Parameters

type[in] The type of limit.

Value	Description
0	No limits
1	duration limit in seconds.
2	file size limit in MB.

value[in] The value of limit. If the value is 0, it means no limit on each media file. When a media file reaches the limit, the control finishes processing the file and creates a new media file.

Return Value

The limit value same as the input if the method call is successful. Otherwise, the current value returned.

Remarks

None.

4.7. StartRecord()

Start to record the media streaming into a specified file.

Definition

BOOL StartRecord(BSTR *filename*)

Parameters

filename[in] The filename of the recording file. If the value is NULL or 「 」, the filename is specified by the control according to MediaFilePath and MediaFilePrefix.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.8. Stop()

Disconnect the remote device.

Definition

BOOL Stop()

Parameters

None.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.9. StopRecord()

Stop recording.

Definition

BOOL StopRecord()

Parameters

None.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.10. ZoomDigitally()

Digitally zoom the video in the center of a specified coordinate.

Definition

LONG zoomDigitally(LONG *zoom*, LONG *x*, LONG *y*)

Parameters

zoom[in] The magnitude of zoom. From 100 (1x) to 1600 (16x).

x[in] The zoom center position in x-coordinate.

y[in] The zoom center position in y-coordinate.

Return Value

The zoom magnitude same as the input if the method call is successful. Otherwise, the current magnitude returned.

Remarks

None



4.11. GetPort()

Get protocol port number.

Definition

LONG GetPort(long type)

Parameters

type[in] protocol

Value	Description
0	RTSP
1	HTTP

Return Value

Return port number.

Remarks

None.

4.12. SetPort()

Set protocol port number.

Definition

LONG SetPort(long type, long val)

Parameters

type[in] protocol

Value	Description
0	RTSP
1	HTTP

val[in] port number 0-65535

Return Value

Return the port number.

Remarks

None.





4.13. **GetMirror()**

Get mirror status.

Definition

BOOL GetMirror(long type)

Parameters

type[in] long

Value	Description
1	MIRROR_HORIZONTAL
2	MIRROR_VERTICAL

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.14. SetMirror()

Set mirror status.

Definition

BOOL SetMirror([in] long type, [in] VARIANT_BOOL val)

Parameters

type[in] long

Value	Description
1	MIRROR_HORIZONTAL
2	MIRROR_VERTICAL

val[in] VARIANT_BOOL

Value	Description
True	Enable
False	Disable

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.15. ShowFileFolderDialog()

Show file folder dialog.

Definition

BSTR ShowFileFolderDialog(BSTR def_path, long style)

Parameters

def_path[in] Default path.

style[in] unused.

Return Value

Get select folder path.

Remarks

None.

4.16. SetMDArea()

Set motion detection area.

Definition

BOOL SetMDArea([in] long area, [in] long x, [in] long y, [in] long w, [in] long h)

Parameters

area[in] 1-3

x[in] left-top coordinates x

y[in] left-top coordinates y

w[in] width.

h[in] height

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.17. GetMDAreaX()

Get motion detection left-top coordinates x.

Definition

LONG GetMDAreaX([in] long area)

Parameters

area[in] area ID 1-3

Return Value

Return left-top coordinates x

Remarks

None.

4.18. GetMDAreaY()

Get motion detection left-top coordinates y.

Definition

LONG GetMDAreaY([in] long area)

Parameters

area[in] area ID 1-3

Return Value

Return left-top coordinates y

Remarks

None.



4.19. GetMDAreaWidth()

Get motion detection area width.

Definition

LONG GetMDAreaWidth(long area)

Parameters

area[in] 1-3

Return Value

Return area width.

Remarks

None.

4.20. GetMDAreaHeight()

Get motion detection height.

Definition

LONG GetMDAreaHeight(long area)

Parameters

area[in] 1-3

Return Value

Return area height.

Remarks

None.



4.21. ControlDecoding()

Set decoding mode.

Definition

BOOL ControlDecoding([in] long arg, [in] long arg2)

Parameters

arg[in]

Value	Description
10901	NORMAL
9901	I_ONLY

arg2[in] decoding Time

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.22. LoadClientCfg()

Get client config value saved in client_cfg.ini.

Definition

BSTR LoadClientCfg([in] BSTR name)

Parameters

name[in] INI key name.

Return Value

Get key name value.

Remarks

None.



4.23. SaveClientCfg()

Set client config value in client_cfg.ini.

Definition

BOOL SaveClientCfg([in] BSTR name)

Parameters

name[in] INI key name.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.24. ApplyAllSavedClientCfg()

Apply all saved client configurations.

Definition

BOOL ApplyAllSavedClientCfg()

Parameters

None

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

MediaType, MediaFilePath, MediaFilePrefix, SnapshotFilePath, SnapshotFilePrefix will be applied at once.



4.25. GetRGB()

Get coordinates one pixel RGB value.

Definition

LONG GetRGB([in] long x, [in] long y)

Parameters

x[in] coordinates X

y[in] coordinates Y

Return Value

Return RGB value.

Remarks

None.

4.26. GetVideoRenderInfo()

Definition

BSTR GetVideoRenderInfo()

Parameters

None.

Return Value

Return string "VideoWidth || VideoHeight || VideoRenderMode || VideoRenderRate."

Remarks

None.



4.27. *MouseWheel()*

Send OnMouseWheel message to notify ActiveX to do digital zoom.

Definition

BOOL MouseWheel(long key, long delta, long x, long y)

Parameters

key[in]

delta[in] Indicates mouse wheel forward/backward in +/- scale units.

x[in] The mouse position in the x-coordinate.

y[in] The mouse position in the y-coordinate.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.28. *SetZoomInLimit()*

Set zoom in limit.

Definition

BOOL SetZoomInLimit(long limit)

Parameters

limit[in] 100(1X)-1600(16X)

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.29. GetZoomInLimit()

Get zoom in limit.

Definition

LONG GetZoomInLimit()

Parameters

None.

Return Value

Return zoom in limit value.

Remarks

None.

4.30. SetBackground()

Stop rendering video to reduce CPU load.

Definition

BOOL SetBackground([in] VARIANT_BOOL _bg)

Parameters

_bg[in] TRUE/FALSE.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.31. **SetFullScreenEnabled()**

Switch to fullscreen directory or fire a full screen event instead.

Definition

BOOL SetFullScreenEnabled([in] VARIANT_BOOL full)

Parameters

full[in]

Value	Description
True	Fullscreen
False	None Fullscreen

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

Enabled by default.



4.32. *SetRenderFPSFactor()*

Divide the render FPS by the factor.

Definition

BOOL SetRenderFPSFactor([in] long factor)

Parameters

factor[in] ranges form 1 to 5.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

Default is 1.

4.33. *GetCurrentImage()*

Get current display image buffer (max 5 frames every minute).

Definition

BOOL GetCurrentImage([in] long theFormat, [in] VARIANT* theBuffer, [in] long* theBufferSize)

Parameters

theFormat[in] ranges form 0 to 1.

Value	Description
0	.JPG
1	.BMP

theBuffer[in] get the image buffer point. (theBuffer.pbVal)

theBufferSize[in] get the buffer size in bytes.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

Do not release the return buffer.

If you want to get every frame then use this method at OnNewImage() Event.

4.34. *SetAudioTransmitData()*

Set upload audio data.

Definition

BOOL SetAudioTransmitData([in] VARIANT* theBuffer)

Parameters

theBuffer [in] Set upload audio buffer point. (theBuffer.pbVal)

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None

4.35. *SetDisplayText()*

Set display text.

Definition

BOOL SetDisplayText ([in] BSTR szText, [in] long R, [in] long G, [in] long B)

Parameters

szText [in] Text.

R [in] = color red.

G [in] = color green.

B [in] = color blue.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None

4.36. SaveCurrentImageEx()

Save the frame into an image file by time.

Definition

BOOL SaveCurrentImage(int *format*, BSTR *filename*, long *sec*, long *usec*)

Parameters

format[in] The image format.

Value	Description
0	JPEG.

filename[in] The filename of the saved iamge.

sec[in] PTS time, get from OnEventEx.

usec[in] PTS time get from OnEventEx.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

4.37. StartRecordEx()

Start to record the media streaming into a specified file by time.

Definition

BOOL StartRecord(BSTR *filename*, long *sec*, long *usec*)

Parameters

filename[in] The filename of the recording file. If the value is NULL or 「 」, the filename is specified by the control according to MediaFilePath and MediaFilePrefix.

sec[in] PTS time, get from OnEventEx.

usec[in] PTS time get from OnEventEx.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.



4.38. StopRecordEx()

Stop recording by time.

Definition

BOOL StopRecord(long *sec*, long *usec*)

Parameters

sec[in] PTS time, get from OnEventEx.

usec[in] PTS time get from OnEventEx.

Return Value

TRUE if the method call is successful. Otherwise, FALSE returned.

Remarks

None.

5. Events

5.1.OnClick()

When a user clicks a mouse button, this event is triggered.

Definition

OnClick(LONG *x*, LONG *y*)

Parameters

x[in] The mouse position in the x-coordinate.

y[in] The mouse position in the y-coordinate.

Remarks

None.

5.2. OnError()

When there is an error occurred, this event is triggered.

Definition

OnError(LONG *level*, BSTR *msg*)

Parameters

level[in] The error level.

Value	Description
0	Error.
1	Warning.
2	Info.
3	Debug.

msg[in] The error message.

Remarks

None.



5.3. OnMediaInfo()

Every second, this event is triggered to provide the media information.

Definition

OnMediaInfo(BSTR *info*)

Parameters

info[in] The media information which is a string consisting of several string segments. Each segment separated by a double pipeline is value of an entry of information like bit rate, frame rate and so on. The format: is as follows:

Image Width||Image Height||Frame Rate||Bit Rate (Kbps)

Ex:

640||480||30||256

Remarks

None.

5.4. OnMouseDown()

When a user clicks a mouse button down, this event is triggered.

Definition

OnMouseDown(LONG *button*, LONG *key*, LONG *x*, LONG *y*)

Parameters

button[in] The mouse button clicked.

Value	Description
1	The left mouse button is clicked.
2	The right mouse button is clicked.
16	The middle mouse button is clicked.

key[in] The SHIFT or CTRL key pressed.

Value	Description
4	The shift key is pressed.
8	The control key is pressed.

x[in] The mouse position in the x-coordinate.

y[in] The mouse position in the y-coordinate.

Remarks

None.



5.5. OnMouseMove()

When a user moves a mouse, this event is triggered.

Definition

OnMouseMove(LONG *button*, LONG *key*, LONG *x*, LONG *y*)

Parameters

button[in] The mouse button clicked. The values can be referred to OnMouseDown() event.

key[in] The SHIFT or CTRL key pressed. The values can be referred to OnMouseDown() event.

x[in] The mouse position in the x-coordinate.

y[in] The mouse position in the y-coordinate.

Remarks

None.

5.6. OnMouseUp()

When a user releases a mouse button up, this event is triggered.

Definition

OnMouseUp(LONG *button*, LONG *key*, LONG *x*, LONG *y*)

Parameters

button[in] The mouse button clicked. The values can be referred to OnMouseDown() event.

key[in] The SHIFT or CTRL key pressed. The values can be referred to OnMouseDown() event.

x[in] The mouse position in the x-coordinate.

y[in] The mouse position in the y-coordinate.

Remarks

None.

5.7. OnRequestResponded()

When the server responds a request, this event is triggered.

Definition

OnRequestResponded(LONG *uid*, LONG *code*, BSTR *body*)

Parameters

uid[in] The unique ID of the request responded.

code[in] The response code of the request. Refer to HTTP response code.

Value	Description
200	OK.
Otherwise	Failed.

body[in] The response body in text.

Remarks

None.

5.8. OnStatusChange()

When the status of the control is changed, this event is triggered.

Definition

OnStatusChange(LONG *curr*, LONG *prev*)

Parameters

curr[in] The current status of the control.

prev[in] The previous status of the control.

Remarks

None.



5.9. OnDoubleClick()

When a user double clicks a mouse button, this event is triggered.

Definition

OnDoubleClick(LONG *x*, LONG *y*)

Parameters

x[in] The mouse position in the x-coordinate.

y[in] The mouse position in the y-coordinate.

Remarks

None.

5.10. OnMouseWheel()

When a user rolls a mouse wheel, this event is triggered.

Definition

OnMouseWheel(long *key*, long *delta*, long *x*, long *y*)

Parameters

key[in]

delta[in] Indicates distance rotated.

x[in] The mouse position in the x-coordinate

y[in] The mouse position in the y-coordinate.

Remarks

None.



5.11. OnEvent()

When an event fire, this event is triggered.

Definition

OnEvent(BSTR event)

Parameters

event[in] Event name

Remarks

EventName||EventType||(HWinputs_1)(HWinputs_2)(HWinputs_3)(HWinputs_4)||SWInput_src||SWInput_channel||SWInput_status||year||mon||mday||hour||min||sec||ms

EventName(could be none)

name

EventType

EMM_EVENT_TYPE_SCHED = 'S',
EMM_EVENT_TYPE_TRIGGER = 'T',
EMM_EVENT_TYPE_FAKE = 'F'

HWinputs_X(none - 4)

EMM_EVENT_DO_NOT_CARE = 'x',
EMM_EVENT_TRIGGER_ON_INACTIVATION = '0', // LOW | '0'
EMM_EVENT_TRIGGER_ON_ACTIVATION = '1', // HIGH | '1'
EMM_EVENT_TRIGGER_ON_CHANGE = '2'

SWInput_src

EMM_EVENT_TRIGGER_SRC_NULL,
EMM_EVENT_TRIGGER_SRC_BOOT,
EMM_EVENT_TRIGGER_SRC_VIDEO,
EMM_EVENT_TRIGGER_SRC_MD,
EMM_EVENT_TRIGGER_SRC_IO,
EMM_EVENT_TRIGGER_SRC_BD,

SWInput_channel

channel number





SWInput_status

```
EMM_EVENT_DO_NOT_CARE = 'x',  
EMM_EVENT_TRIGGER_ON_INACTIVATION = '0', // LOW | '\'  
EMM_EVENT_TRIGGER_ON_ACTIVATION = '1',  // HIGH | '/'  
EMM_EVENT_TRIGGER_ON_CHANGE = '2'
```

The time is UTC time.

5.12. OnReconnect()

When reconnect happend, this event is triggerd.

Definition

OnReconnect(long type, VARIANT_BOOL multicast)

Parameters

type[in] protocol type

Value	Description
1	UDP
2	RTSP
3	HTTP

multicast[in] Is multicast?

Remarks

None.



5.13. OnMediaRecorded()

When a media file is recorded, this event is triggered.

Definition

OnMediaRecorded([in] BSTR path)

Parameters

path[in] media file path.

Remarks

None.

5.14. OnFullScreen()

When full screen is disabled and pop up menu item is click, this event is triggered.

Definition

OnFullScreen()

Parameters

None

Remarks

User should catch the event to simulate full screen effects.

5.15. OnRecord()

When user right-click popup record btn, this event is triggered.

Definition

OnRecord([in]long status)

Parameters

status[in] result.

Remarks

user can write code when record btn click.

5.16. OnSnapshot()

When user right-click popup snapshot btn, this event is triggered.

Definition

OnSnapshot([in]long status)

Parameters

None.

Remarks

user can write code when snapshot btn click.

5.17. OnNewAudioFrame()

Get audio WAVEHDR header point before play.

Definition

OnNewAudioFrame([in] VARIANT theBuffer)

Parameters

theBuffer[in] frame buffer point.

Remarks

Do not release frame buffer.
the buffer point is WAVEHDR point.

```
typedef struct wavehdr_tag {  
    LPSTR    lpData;           /* pointer to locked data buffer */  
    DWORD    dwBufferLength;   /* length of data buffer */  
    DWORD    dwBytesRecorded;  /* used for input only */  
    DWORD_PTR dwUser;          /* for client's use */  
    DWORD    dwFlags;          /* assorted flags (see defines) */  
    DWORD    dwLoops;          /* loop control counter */  
    struct wavehdr_tag FAR *lpNext; /* reserved for driver */  
    DWORD_PTR reserved;        /* reserved for driver */  
} WAVEHDR, *PWAVEHDR, NEAR *NPWAVEHDR, FAR *LPWAVEHDR;
```

theBuffer.pbVal is the point.



5.18. OnNewVideoFrame()

Get video decoded buffer before display.

Definition

OnNewVideoFrame([in] VARIANT theBuffer, [in] long theWidth, [in] long theHeight, [in] long thebpp, [in] long theFormat)

Parameters

theBuffer[in] frame buffer point.

theWidth[in] video frame width.

theHeight[in] video frame height.

thebpp[in] pixel format , ex. 24bit or 32bit.

theFormat[in] 0:RGB 1:YUV

Remarks

Do not release frame buffer.
theBuffer.pbVal is the point.

5.19. OnNewEventFrame()

Get event status.

Definition

OnNewEventFrame([in] sEvent)

Parameters

sEvent[in] event string.

Remarks

same OnEvent().

5.20. OnNewEventFrameEx()

Get event status.

Definition

OnNewEventFrame([in] VARIANT theEvent)

Parameters

theEvent[in] event struct.

Remarks

struct TM

```
{
    int tm_sec;          /* Seconds.  [0-60] (1 leap second) */
    int tm_min;          /* Minutes.  [0-59] */
    int tm_hour;         /* Hours.    [0-23] */
    int tm_mday;         /* Day.     [1-31] */
    int tm_mon;          /* Month.   [0-11] */
    int tm_year;         /* Year    - 1900. */
    int tm_wday;         /* Day of week. [0-6] */
    int tm_yday;         /* Days in year.[0-365] */
    int tm_isdst;        /* DST.      [-1/0/1]*/
```

```
#ifdef __USE_BSD
```

```
    long int tm_gmtoff;    /* Seconds east of UTC. */
    char *tm_zone;        /* Timezone abbreviation. */
```

```
#else
```

```
    long int __tm_gmtoff;  /* Seconds east of UTC. */
    char *__tm_zone;      /* Timezone abbreviation. */
```

```
#endif
```

```
};
```

```
/* Configuration Structures */
```

```
typedef enum tagEMM_EVENT_OPCODE {
    EMM_EVENT_OP_CREATE,
    EMM_EVENT_OP_MODIFY,
    EMM_EVENT_OP_REMOVE,
    EMM_EVENT_OP_BYPASS
} EMM_EVENT_OPCODE;
```

```
typedef enum tagEMM_EVENT_TYPE {
    EMM_EVENT_TYPE_SCHED = 'S',
    EMM_EVENT_TYPE_TRIGGER = 'T',
    EMM_EVENT_TYPE_FAKE = 'F'
} EMM_EVENT_TYPE;
```



```
typedef enum tagEMM_EVENT_TRIGGER_STATUS {  
    EMM_EVENT_DO_NOT_CARE = 'x',  
    EMM_EVENT_TRIGGER_ON_INACTIVATION = '0', // LOW | '\'  
    EMM_EVENT_TRIGGER_ON_ACTIVATION = '1',  // HIGH | '/'  
    EMM_EVENT_TRIGGER_ON_CHANGE = '2'  
} EMM_EVENT_TRIGGER_STATUS;
```

```
typedef enum tagEMM_EVENT_TRIGGER_SRC {  
    EMM_EVENT_TRIGGER_SRC_NULL,  
    EMM_EVENT_TRIGGER_SRC_BOOT,  
    EMM_EVENT_TRIGGER_SRC_VIDEO,  
    EMM_EVENT_TRIGGER_SRC_MD,  
    EMM_EVENT_TRIGGER_SRC_IO,  
    EMM_EVENT_TRIGGER_SRC_BD,  
} EMM_EVENT_TRIGGER_SRC;
```

```
typedef enum tagEMM_EVENT_TRIGGER_CHANNEL {  
    EMM_EVENT_TRIGGER_CHANNEL_NULL = -1,  
    EMM_EVENT_TRIGGER_CHANNEL_1 = 0,  
    EMM_EVENT_TRIGGER_CHANNEL_2,  
    EMM_EVENT_TRIGGER_CHANNEL_3,  
    EMM_EVENT_TRIGGER_CHANNEL_4,  
    EMM_EVENT_TRIGGER_CHANNEL_ANY,  
    EMM_EVENT_TRIGGER_CHANNEL_ALL  
} EMM_EVENT_TRIGGER_CHANNEL;
```

```
typedef struct tagConfig_Event {  
    EMM_EVENT_OPCODE OP;  
  
    unsigned Index;  
    EMM_EVENT_TYPE    Type;    // S,T  
    char    Name[32];  
  
    BOOL    Enabled;  
    BOOL    Active;  
    unsigned Priority;  
  
    EMM_EVENT_TRIGGER_STATUS HWInputs[4];  
  
    struct SWINPUT {  
        EMM_EVENT_TRIGGER_SRC    src;  
        EMM_EVENT_TRIGGER_CHANNEL channel;  
        EMM_EVENT_TRIGGER_STATUS status;  
    } SWInput;
```



```
BOOL Weekdays[8];

struct TM Starttime;    // tm_hour [0-23] and tm_min [0-59]
struct TM Duration;

int Image;
BOOL IncludePreTrigger;
unsigned PreTriggerSize;    // number of frames
unsigned PreTriggerInterval; // ms

BOOL IncludePostTrigger;
unsigned PostTriggerSize;    // number of frames
unsigned PostTriggerInterval; // ms

BOOL IncludeBestEffort;
unsigned BestEffortDuration; // seconds
unsigned BestEffortInterval; // ms

BOOL IncludeBatch;
double BatchSize;    // seconds

char FileName[32];    // base file name for uploaded file
char Suffix[32];    // wildcard pattern
unsigned MaxSequenceNumber;

struct timeval AbsTime;    // absolute time
int TimerId;
} EMM_CONFIG_EVENT, *LPEMM_CONFIG_EVENT;
```

5.21. OnEventEx()

Get Software event.

Definition

OnEventEx([in] long Type, [in] long Status, [in] BSTR sHWInput, [in] long sec, [in] long usec)

Parameters

Type[in] reference EMM_EVENT_TYPE.

Status[in] reference EMM_EVENT_TRIGGER_STATUS.

sHWInput[in] Hardware input.

sec[in] PTS time.

usec[in] PTS time.

Remarks

None.

5.22. OnRTPAudioFrame()

Get pure Audio frame data.

Definition

OnRTPAudioFrame([in] VARIANT theBuffer, [in] long iSize, [in] long iPTS_sec, [in] long iPTS_usec)

Parameters

theBuffer[in] data point.

iSize[in] data size.

iPTS_sec[in] PTS sec.

iPTS_usec[in] PTS usec.

Remarks

None.



5.23. OnRTPVideoFrame()

Get pure Video frame data.

(Can get H.264 raw data which contain I and P frames.)

Definition

OnRTPVideoFrame([in] VARIANT theBuffer, [in] long iSize, [in] long iPTS_sec, [in] long iPTS_usec, [in] long iType)

Parameters

theBuffer[in] data point.

iSize[in] data size.

iPTS_sec[in] PTS sec.

iPTS_usec[in] PTS usec.

iType[in] key frame.

Remarks

OnRTPVideoFrame can get H.264 raw data which contain I and P frames.

OnNewImage is a callback function which will pass JPEG image data ;

you can use GetCurrentImage() to get JPEG data (max 5 frames every minute).

You may use OnRTPVideoFrame to decode and display real time h.264 streaming by your own H.264 decoder. Or you may use OnNewImage to display streaming with JPEG image if you are not familiar with H.264 decode process.



5.23. OnRTPEventFrame()

Get pure Event frame data.

Definition

OnRTPEventFrame([in] VARIANT theBuffer, [in] long iSize, [in] long iPTS_sec, [in] long iPTS_usec)

Parameters

theBuffer[in] data point.

iSize[in] data size.

iPTS_sec[in] PTS sec.

iPTS_usec[in] PTS usec.

Remarks

Reference 5.20. OnNewEventFrameEx()