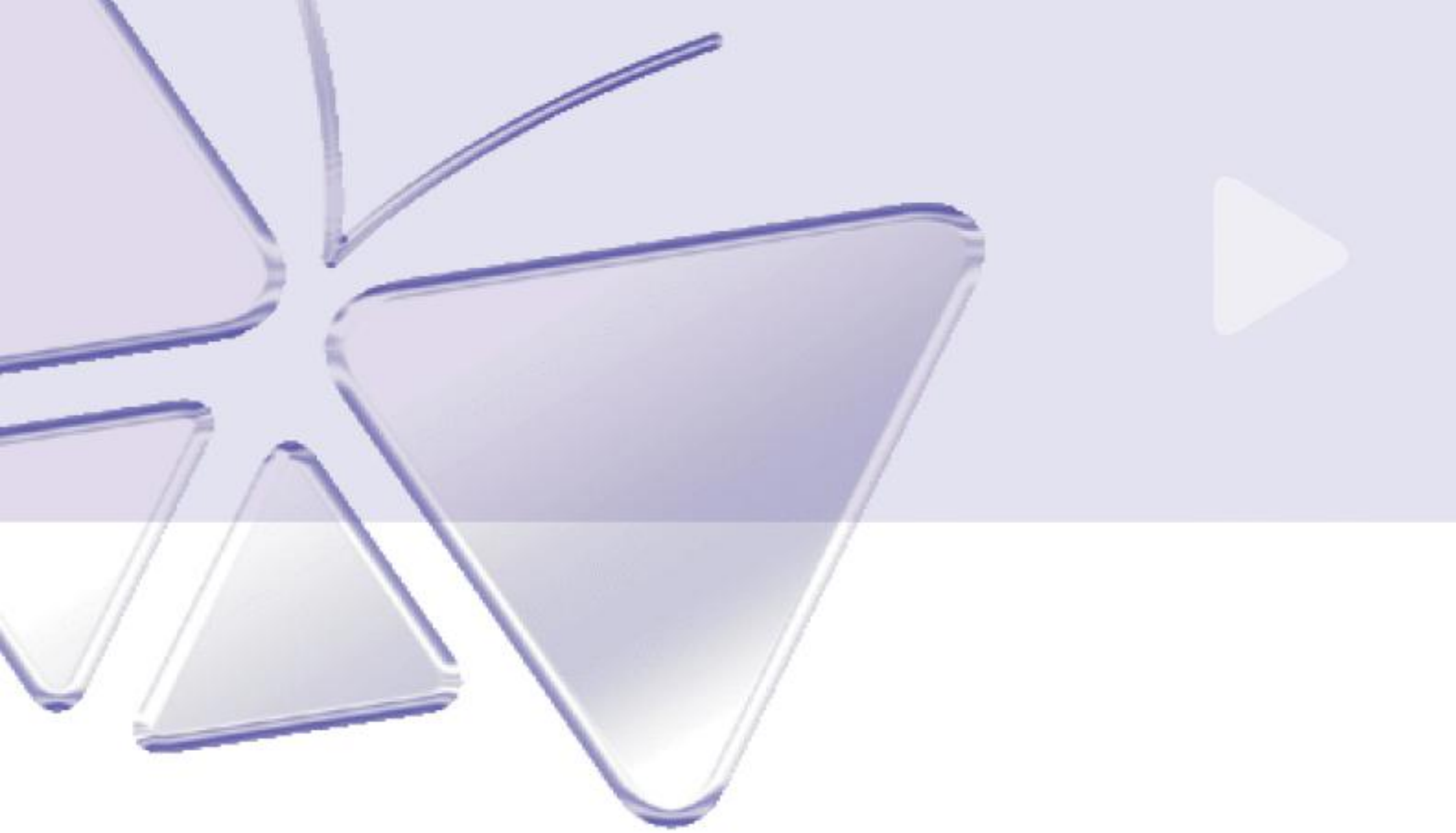

ACTi SDK-10000

ActiveX Control Edition

V1.2.90

API Reference Guide

www.acti.com

ACTi SDK-10000

V1.2 Edition Jul, 2010

Table of Contents

1 Introduction

Streaming Control is developed using Microsoft ATL (Active Template Library) v3.0 frameworks. By applying ATL, Streaming Control has better performance with smaller executable size.

Benefits of Streaming Control

The benefit of streaming control is listed as follow.

- **Object-oriented design**: With streaming control, you may benefit from pure object-oriented programming. There are six major objects in streaming control, they are:
- **Multiple programming Supports**: Streaming control is developed standard ActiveX Control framework, hence, it can be programmed with Microsoft Visual C++, Microsoft Visual Basic, and ASP with VBScript and HTML with Java Script.
- **Encapsulate the complexity of streaming architecture**: With streaming control, user may skips the trivial details of streaming architecture, such as TCP, UDP, RTP, Multicast, etc; instead, with simple methods, users may focus on developing business rules for applications.
- **Integrates and adopts multiple media formats**: streaming control adopts streaming and file media formats.

Streaming Objects

Streaming control consists of six major objects, they are listed as follow:

Name	*Description*
`DIO`	Controls digital input and digital output.
`Media`	Media manipulation. Media object can be streaming media or file media.
`MotionDetection`	Motion detection object handles all necessary operations when an event is triggered.
`Network`	Network object controls the streaming parameters, with TCP, UDP or Multicast protocols.
`PTZ`	PTZ object handles all operations on pan, tilt and zoom events.
`UI`	UI object handles the interaction between the streaming object and the user's application.

Streaming Control Architecture

Streaming control can be programmed using Visual BASIC, Visual C++, Visual Studio.Net and HTML + JavaScript, and the calling conventions are the same through various developing environment.

By adopting DirectX platform, Streaming Control is able to render standard MPEG-4 streaming file and supports other video formats in the future.

2 Control Property

This section lists media property. These values have to be defined before invoking Control methods.

Media Property

Name	*Description*
AutoReconnect	**[I/O]** Sets/Gets whether certain streaming media reconnect automatically when connection lost
AutoReconnectInterval	**[I/O]** Sets/Gets reconnect interval whether certain streaming media reconnect automatically when connection loss
ASEControlPort	**[I/O]** Sets/Gets the special control port for streaming engine
ASEMediaSource	**[I/O]** Identifies the source of the media stream for streaming engine
錯誤! 找不到參照來源。	**[I/O]** Sets the user name of the media stream for streaming engine
ASEMediaPassword	**[I]** Sets the password to access the media stream for streaming engine
ASEStreamingPort	**[I/O]** Sets/Gets the streaming port for streaming port
BeginTime	**[O]** The time stamp in the playback media
BitRate	**[I/O]** Sets/Gets the bit-rate value of video configuration
Brightness	**[I/O]** Sets/Gets the brightness value of video configuration
CodecType	**[I/O]** Sets/Gets the codec type.
ContentStatus	**[O]** Check current streaming status
Contrast	**[I/O]** Sets/Gets the contrast value of video configuration
ControlPort	**[I/O]** Sets/Gets the special control port
DeviceChipType	**[O]** Gets the device chip type
Duration	**[O]** Total playing time of current playback media
EnableASE	**[I/O]** Connect to IP device by streaming engine or not
EnableBorder	**[I/O]** Enable ActiveX control border
EndTime	**[O]** The time stamp in the playback media
Fps	**[I/O]** Sets/Gets the frame per second value of video configuration
FrameRateMode	**[O]** Returns current camera frame rate mode type
HttpPort	**[I/O]** Sets the special HTTP port
Hue	**[I/O]** Sets/Gets the hue value of video configuration
ID	**[I/O]** Sets/Gets the camera ID
錯誤! 找不到參照來源。	**[O]** Check if connected device is a quad device
LastRecFileSize	**[O]**Gets the last record file size
longLastRecFileSize	**[O]**Gets the last record file size in long
MediaChannel	**[I/O]** Sets/Gets the camera channel ID
MediaPassword	**[I]** Sets the password to access the media stream

MediaSource	**[I/O]** Identifies the source of the media stream
MediaType	**[I/O]** Sets/Gets the media type: Preview, Playback or PTZ status
MediaUsername	**[I/O]** Sets the user name of the media stream
MulticastIP	**[I/O]** Sets/Gets the multicast IP address
MulticastPort	**[I/O]** Sets/Gets the multicast port
Mute	**[I]** Turn on or off the PC sound device.
NetworkStatus	**[O]** Check if current network status
PreRecordTime	**[I/O]** Sets/Gets the pre-recording time before an event
PostRecordingTime	**[I/O]** Sets/Gets the post-recording time after an event
RecordType	**[O]** Returns current camera recording status
RegisterPort	**[I/O]** Gets/Sets the register port
ReplaceTimeCodeByLocalTime	**[I]** Replace time code by local time
Resolution	**[I/O]** Sets/Gets the resolution value of video configuration
RTSPPort	**[I/O]** Sets the special RTSP port
Saturation	**[I/O]** Sets/Gets the saturation value of video configuration
StreamingPort	**[I/O]** Sets/Gets the streaming port
StreamType	**[I/O]** Sets/Gets the streaming type
strLastRecFileSize	**[O]** Gets the last record file size in string
VariableF	**[I/O]** Sets/Gets the variable frame per second value of video configuration
Version	**[O]** Returns the SDK version
Volume	**[I/O]** Sets/Gets the PC sound Volume**.**
DeviceType	**[I/O]** Sets/Gets the device type value.
QuadDeviceMode	**[I/O]** Sets/Gets device mode value on quad device.

AutoReconnect

Description

`[I/O]` Sets/Gets whether certain streaming media reconnect automatically when connection lost

Syntax

```
AutoReconnect = Input_Value
Return_Value = AutoReconnect [ default: 0 ]
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets the value of AutoStart flag
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>AutoReconnect sets to no</td></tr>
<tr><td><b>1</b></td><td>AutoReconnect sets to yes</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Input_Value	LONG	Returns current settings value

Remarks

Default is no(0). Meaning that when network connection is broken, SDK will not re-connect the session automatically

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100”;
ObjStreaming.MediaUsername = “Admin”;
ObjStreaming.MediaPassword = “123456”;
objStreaming.AutoReconnect = 1;
objStreaming.Connect(0) ;
objStreaming.StartStream();
If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

`Connect(), Play()`

AutoReconnectInterval

Description

[I/O] Sets/Gets auto reconnect interval whether certain streaming media reconnect automatically when connection loss

Syntax

```
AutoReconnectInterval = Input_Value
Return_Value = AutoReconnectInterval [ default: 10 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the value of interval

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns the value of interval

Remarks

Default is 10 second with reconnect interval.

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100”;
ObjStreaming.MediaUsername = “Admin”;
ObjStreaming.MediaPassword = “123456”;
objStreaming.AutoReconnect = 1;
objStreaming.AutoReconnectInterval = 10; // 10 second will try to reconnect again
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

Connect(), Play()

ASEControlPort

Description

`[I/O]` Sets/Gets the special control port for streaming engine

Syntax

```
ASEControlPort = Input_Value
Return_Value = ASEControlPort
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special control port for streaming engine

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value for streaming engine

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;
... ...
ObjStreaming.ASEControlPort = 6001
... ...
objStreaming.Connect(0) ;
objStreaming.StartStream();

......
```

See Also

`RegisterPort, StreamingPort, MulticastPort, HttpPort`

ASEMediaSource

Description

[I/O] Sets/Gets the source of the media stream for streaming engine.

Syntax

```
ASEMediaSource = "Input_Value"
"Return_Value" = ASEMediaSource
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the source of the media stream The source can be an IP address or hostname

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Returns current value stored in ASEMediaSource.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = "192.168.0.100" ;
ObjStreaming.MediaUsername = "Admin" ;
ObjStreaming.MediaPassword = "123456" ;

ObjStreaming.ASEMediaSource = "172.16.1.178" ; // Streaming engine server address
ObjStreaming.ASEMediaUserName = "Admin" ;
ObjStreaming.ASEMediaPassword = "123456" ;
ObjStreaming.EnableASE = TRUE ;


objStreaming.Connect(0) ;
objStreaming.StartStream();
```

```
If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

Connect()

ASEMediaUserName

Description

[I/O] Sets the user name of the media stream for streaming engine

Syntax

```
ASEMediaUserName = “Input_Value”
“Return_Value” = ASEMediaUserName
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the user name of the media stream for streaming engine

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the user name of the media stream for streaming engine

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;

ObjStreaming.ASEMediaSource = “172.16.1.178” ; // Streaming engine server address
ObjStreaming.ASEMediaUserName = “Admin” ;
ObjStreaming.ASEMediaPassword = “123456” ;
ObjStreaming.EnableASE = TRUE ;


objStreaming.Connect(0) ;
objStreaming.StartStream();
```

```
If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

Connect()

ASEMediaPassword

Description

`[I/O]` Sets/Gets the password of the media stream for streaming engine

Syntax

`ASEMediaPassword = “`*`Input_Value`*`”`

`“Return_Value”` `= ASEMediaPassword`

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the password of the media stream for streaming engine

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the password of the media stream for streaming engine

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;

ObjStreaming.ASEMediaSource = “172.16.1.178” ; // Streaming engine server address
ObjStreaming.ASEMediaUserName = “Admin” ;
ObjStreaming.ASEMediaPassword = “123456” ;
ObjStreaming.EnableASE = TRUE ;


objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

`Connect()`

ASEStreamingPort

Description

`[I/O]` Sets/Gets the streaming port for streaming engine

Syntax

```
ASEStreamingPort = Input_Value
Return_Value = ASEStreamingPort
```

Parameters

Name	*Type*	*Description*
`Input_Value`	`LONG`	Set the special streaming port for streaming engine

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	Returns current settings value for streaming engine

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;
......
ObjStreaming.ASEStreamingPort = 6002 ;
......
objStreaming.Connect(0) ;
objStreaming.StartStream();
......
```

See Also

`ControlPort, MulticastPort, HttpPort, MulticastPortRegisterPort, StreamingPort`

BeginTime

Description

[o] Gets the time stamp in the playback media

Syntax

Return_Value = BeginTime

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns the time stamp in the playback media

Remarks

NONE

Example

```
objStreaming.MediaType = 2 ;
objStreaming.MediaSource = “C:\abc.raw” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

dwBeginTime = objStreaming.BeginTime ;
```

See Also

Play(), Duration,

BitRate

Description

`[I/O]` Sets/Gets the bit-rate value of video configuration

Syntax

```
BitRate = Input_Value
Return_Value = BitRate
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the bit-rate value of the video configuration

Value	Description
0	28 Kbps
1	56 Kbps
2	128 Kbps
3	256 Kbps
4	384 Kbps
5	500 Kbps
6	750 Kbps
7	1 Mbps
8	1.2 Mbps
9	1.5 Mbps
10	2 Mbps
11	2.5 Mbps
12	3 Mbps
13	3.5 Mbps
14	4 Mbps
15	4.5 Mbps
16	5 Mbps
17	5.5 Mbps
18	6 Mbps

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns the bit-rate value of the video configuration

Remarks

From 3.5M to 6M are available in PaltformT encoders only.

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}

dwBitRate = objStreaming.BitRate    ‘ get current bit rate setting
```

See Also

Resolution, Fps

Brightness

Description

[I/O] Sets/Gets the brightness value of video configuration

Syntax

```
Brightness = Input_Value
Return_Value = Brightness
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets the brightness value of the video configuration<table><tr><th>Minimum</th><th>Maximum</th></tr><tr><td>0</td><td>100</td></tr></table></td></tr>
</table>

Returns

Name	Type	Description
Return_Value	LONG	Returns current brightness value in video configuration

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

```
objStreaming.Brightness = 30 ;
dwBrightness = objStreaming.Brightness ;
```

See Also

Hue, Contrast

CodecType

Description

[I/O] Sets/Gets the codec type.

Syntax

```
CodecType = Input_Value
Return_Value = CodecType
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the value of codec type.

value	description
0	XVID
1	FFMPEG
2	PCI51
3	IPP
4	MJPEG
5	H.264

Returns

NO	*Type*	*Description*
Return_Value	LONG	Return current setting value.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    Alert(ObjStreaming.CodecType);
}
```

See Also

MediaType

ContentStatus

Description

[O] Gets current streaming status

Syntax

Return_Value = **ContentStatus** [*default: 0*]

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Check current streaming status
<table>
<tr><th>Value</th><th>Define</th></tr>
<tr><td>0</td><td>_STREAMING_MODE_STOP</td></tr>
<tr><td>1</td><td>_STREAMING_MODE_START</td></tr>
<tr><td>2</td><td>_STREAMING_MODE_PLAY</td></tr>
<tr><td>3</td><td>_STREAMING_MODE_PAUSE</td></tr>
</table></td></tr>
</table>

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;

If (ObjStreaming.NetworkStatus != 0)
{
    ContentStatus = objStreaming.ContentStatus;
    if(ContentStatus != _STREAMING_MODE_START)
        objStreaming.StartStream();

    objStreaming.Play() ;
}
```

See Also

Contrast

Description

[I/O] Sets/Gets the contrast value of video configuration

Syntax

```
Contrast = Input_Value
Return_Value = Contrast
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the contrast value of the video configuration Minimum: 0 Maximum: 100

Minimum	Maximum
0	100

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current contrast value in video configuration

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

```
objStreaming.Contrast = 30 ;
dwContrast = objStreaming.Contrast ;
```

See Also

Brightness, Hue

ControlPort

Description

`[I/O]` Sets/Gets the special control port

Syntax

```
ControlPort = Input_Value
Return_Value = ControlPort
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special control port

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;

objStreaming.Connect(0) ;
objStreaming.StartStream();

......
```

See Also

`RegisterPort, StreamingPort, MulticastPort, HttpPort`

DeviceChipType

Description

[O] Gets the device chip type

Syntax

Return_Value = DeviceChipType [*default: 0*]

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns the value of device chip type.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.DeviceChipType = 1
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

Duration

Description

`[o]` Gets total playing second time of current playback media

Syntax

`Return_Value` `= Duration`

Parameters

NONE

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	Returns the length of the playing streaming media

Remarks

NONE

Example

```
objStreaming.MediaType = 2 ;
objStreaming.MediaSource = “C:\abc.raw” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

lTotalLength = objStreaming.Duration ;
```

See Also

`BeginTime, Play()`

EnableASE

Description

[I/O] Connect to IP device by streaming engine or not

Syntax

```
EnableASE = Input_Value
Return_Value = EnableASE [ default: FALSE ]
```

Parameters

Name	*Type*	*Description*
Input_Value	Bool	Connect to IP device by streaming engine.

Returns

Name	*Type*	*Description*
Return_Value	Bool	This connection is using streaming engine

Remarks

NONE

Example

```
ObjStreaming.ASEMediaSource = “172.16.1.178” ; // Streaming engine server address
ObjStreaming.ASEMediaUserName = “Admin” ;
ObjStreaming.ASEMediaPassword = “123456” ;
ObjStreaming.EnableASE = TRUE ;
```

See Also

EnableBorder

Description

`[I/O]` Sets/Gets the ActiveX control border status.

Syntax

```
EnableBorder = Input_Value
Return_Value = EnableBorder [ default: FALSE ]
```

Parameters

Name	*Type*	*Description*
Input_Value	BOOL	Set ActiveX control border enabled.

Returns

Name	*Type*	*Description*
Return_Value	BOOL	Get ActiveX control border enabled status

Remarks

NONE

Example

```
ObjStreaming.EnableBorder = TRUE ;
```

See Also

EndTime

Description

`[O]` Gets the time stamp in the playback media

Syntax

`Return_Value` `= EndTime`

Parameters

NONE

Returns

Name	*Type*	*Description*
`Return_Value`	`Long`	Returns the time stamp in the playback media

Remarks

NONE

Example

```
objStreaming.MediaType = 2 ;
objStreaming.MediaSource = “C:\abc.raw” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

dwBeginTime = objStreaming.EndTime ;
```

See Also

`Play(), Duration`

Fps

Description

[I/O] Sets/Gets the frame per second value of video configuration

Syntax

```
Fps = Input_Value
Return_Value = Fps
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the frame per second value of the video configuration Under NTSC mode, the available frame rate are: 30, 6, 3, 1 Under PAL mode, the available frame rate are: 25, 5, 3, 1

⚠ **IMPORTANT:** To set this property, it only works under Variable Frame Rate settings. (Please check Web Configuration setting on this).

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current frame per second value in video configuration

Remarks

If Fps is set to 0, then the video will not be displayed.

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

```
If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}

if (objStreaming.FpsMode == 1) {
    objStreaming.Fps = 30 ;
}
```

See Also

`BitRate, Resolution`

FrameRateMode

Description

`[O]` Gets current camera frame rate mode type

Syntax

`Return_Value` `= FrameRateMode`

Parameters

NONE

Returns

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>Return_Value</em></td><td>LONG</td><td>Returns current camera FPs mode type s
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><strong>1</strong></td><td>Is constant frame rate mode</td></tr>
<tr><td><strong>2</strong></td><td>Is variable frame rate mode</td></tr>
</table>
</td></tr>
</table>

Remarks

Please check Web Configurator (firmware) User's Manual for Variable and Constant Frame Rate settings.

Example

```
......
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

if(objStreaming.FrameRateMode == 0 )
     MessageBox("Constant FPs Mode");
else
     MessageBox("Veriable FPs Mode");
```

See Also

HttpPort

Description

[I/O] Sets/Gets the special Http port

Syntax

```
HttpPort = Input_Value
Return_Value = HttpPort
```

Parameters

Name	*Type*	*Description*
Input_Value	Long	Set the special Http port

Returns

Name	*Type*	*Description*
Return_Value	Long	Returns current settings value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;

objStreaming.Connect(0) ;
objStreaming.StartStream();
......
```

See Also

RegisterPort, StreamingPort, MulticastPort, ControlPort

Hue

Description

[I/O] Sets/Gets the hue value of video configuration

Syntax

```
Hue = Input_Value
Return_Value = Hue
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the hue value of the video configuration Minimum: 0 Maximum: 100

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current hue value in video configuration

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

objStreaming.Hue = 30 ;
dwHue = objStreaming.Hue ;
```

See Also

Contrast, Brightness

ID

Description

`[I/O]` Sets/Gets the camera ID

Syntax

```
ID = Input_Value
Return_Value = ID
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the camera ID Minimum: 1 Maximum: 255

Returns

Name	*Type*	*Description*
Return_Value	LONG	Gets current camera ID

Remarks

This is used in event as callback ID.   For example, motion detection callback needs to assign an ID to identify which video server submit this event.

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;       ‘ ID 1 to identify number 1 of the object
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

LastRecFileSize

Description

[I/O] Sets/Gets the last record file size

Syntax

```
LastRecFileSize = Input_Value
Return_Value = LastRecFileSize
```

Parameters

Name	*Type*	*Description*
Input_Value	ULONGLONG	Sets the last record file size

Returns

Name	*Type*	*Description*
Return_Value	ULONGLONG	Gets the last record file size

Remarks

NONE

Example

```
ObjStreaming.LastRecFileSize;
```

See Also

MediaUsername, Connect()

longLastRecFileSize

Description

`[I/O]` Sets/Gets the last record file size in long

Syntax

```
longLastRecFileSize = Input_Value
Return_Value = longLastRecFileSize
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the last record file size

Returns

Name	*Type*	*Description*
Return_Value	LONG	Gets the last record file size

Remarks

NONE

Example

```
ObjStreaming.longLastRecFileSize;
```

See Also

`MediaUsername, Connect()`

MediaChannel

Description

[I/O] Sets/Gets the camera channel ID

Syntax

```
MediaChannel = Input_Value
Return_Value = MediaChannel
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the camera channel ID

Returns

Name	*Type*	*Description*
Return_Value	LONG	Gets the camera channel ID

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaChannel = 1 ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

MediaUsername, Connect()

MediaPassword

Description

[I] Sets the password to access the media stream

Syntax

`MediaPassword = “Input_Value”`

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the password to access the media source

Returns

NONE

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

MediaUsername, Connect()

MediaSource

Description

`[I/O]` Sets/Gets the source of the media stream

Syntax

`MediaSource = “`*`Input_Value`*`”`

`“Return_Value”`` = MediaSource`

Parameters

Name	*Type*	*Description*
`Input_Value`	`BSTR`	Sets the source of the media stream The source can be an IP address or a filename

Returns

Name	*Type*	*Description*
`Return_Value`	`BSTR`	Returns current value stored in MediaSource.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

`Connect()`

MediaType

Description

`[I/O]` Sets/Gets the media type: Preview, Playback or PTZ status

Syntax

```
MediaType = Input_Value
Return_Value = MediaType [ default: 0 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the value of MediaType flag

Value	Description
0	Preview but no control
1	Preview with Unicast
2	Playback
3	Send PTZ command, but no streaming
4	Preview with Multicast
5	Preview with Multicast, but no control
6	Multiplayback
7	RTSP
8	AHTTP

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current setting value.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

`Play()`

MediaUsername

Description

`[I/O]` Sets/Gets the user name of the media stream

Syntax

```
MediaUsername = “Input_Value”
“Return_Value” = MediaUsername
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the user name of the media stream

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the user name of the media stream

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

`MediaPassword, Connect()`

MulticastIP

Description

`[I/O]` Sets/Gets the multicast IP address

Syntax

`MulticastIP = “Input_Value”`
`“Return_Value” = MulticastIP`

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the multicast IP address of the media stream

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the multicast IP address of the media stream

Remarks

If you set multicast IP, the streaming preview mode will be using multicast mode otherwise will be use unicase mode.

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
objStreaming.MulticastIP = “228.5.6.1” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

`Connect(), MulticastPort`

Mute

Description

`[I/O]` Sets/Gets the audio to turn on or off.

Syntax

`Mute` `=` *`Input_Value`*

`Return_Value` `=` **`Mute`** `[` *`default: 0`* `]`

Parameters

Name	*Type*	*Description*
`Input_Value`	`LONG`	Specifies the flag to decode frames

Value	Description
0	Audio Out
1	Audio Mute

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	Returns the current setting value.

Remarks

NONE

Example

```
......

objStreaming.Play() ;
objStreaming.Mute = 1 ;
objStreaming.SetVolume(30) ;
```

See Also

MulticastPort

Description

`[I/O]` Sets/Gets the multicast port

Syntax

```
MulticastPort = Input_Value
Return_Value = MulticastPort
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special Multicast port

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current setting value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;

objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

`Connect(), MulticastIP`

NetworkStatus

Description

[O] Gets current network status

Syntax

Return_Value = **NetworkStatus** [*default: 0*]

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Check if current status did connected or not
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Disconnected</td></tr>
<tr><td>1</td><td>Conenct control Only</td></tr>
<tr><td>2</td><td>Conenct Streaming</td></tr>
<tr><td>3</td><td>Conenct control and Streaming</td></tr>
<tr><td>4</td><td>Disconnecting</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

PreRecordTime

Description

`[I/O]` Sets/Gets the pre-recording time before an event

Syntax

```
PreRecordTime = Input_Value
Return_Value = PreRecordTime
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets the pre-recording time before an event
<table>
<tr><th>Minimum</th><th>Maximum</th></tr>
<tr><td>5 (Sec)</td><td>30 (Sec)</td></tr>
</table>
</td></tr>
</table>

Returns

Name	Type	Description
Return_Value	LONG	Gets the value of the pre-recording time before an event

Remarks

NONE

Example

```
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (objStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
if (objStreaming.ContentStatus == 1) {
     objStreaming.PreRecordTime = 10 ;
     objStreaming.PostRecordingTime = 10 ;
     objStreaming.StartAlarmRecord(“C:\ABC.raw”);
}
```

See Also

`PostRecordingTime`

PostRecordingTime

Description

`[I/O]` Sets/Gets the post-recording time after an event

Syntax

```
PostRecordingTime = Input_Value
Return_Value = PostRecordingTime
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets the post-recording time after an startalarmrecord
<table>
<tr><th>Minimum</th><th>Maximum</th></tr>
<tr><td>5 (Sec)</td><td>30 (Sec)</td></tr>
</table>
</td></tr>
</table>

Returns

Name	Type	Description
Return_Value	LONG	Returns the value of the post-recording time after an startalarmrecord

Remarks

NONE

Example

```
......
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

if (objStreaming.ContentStatus == 1) {
     objStreaming.PreRecordTime = 10 ;
     objStreaming.PostRecordingTime = 10 ;
     objStreaming.StartAlarmRecord("C:\ABC.raw");
}
```

See Also

`PreRecordTime`

RecordType

Description

`[I/O]` Sets/Gets the current camera recording status

Syntax

```
RecordType = Input_Value
Return_Value = RecordType [ default: 0 ]
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets current camera recording status
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Record Raw file</td></tr>
<tr><td>1</td><td>Record AVI file</td></tr>
<tr><td>2</td><td>Record RAW + IDX file.</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current camera recording status

Remarks

NONE

Example

```
......
objStreaming.Connect(0);
objStreaming.StartStream();
objStreaming.RecordType = 0;
objStreaming.Play() ;

if (objStreaming.ContentStatus == 1) {
     objStreaming.StartRecord(“C:\ABC.raw”);
}
```

See Also

RegisterPort

Description

`[I/O]` Gets/Sets the register port

Syntax

```
RegisterPort = Input_Value
Return_Value = RegisterPort
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special register port

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

`ControlPort, HttpPort, MulticastPort, RegisterPort, StreamingPort`

ReplaceTimeCodeByLocalTime

Description

`[I/O]` Sets/Gets the replace time code by local time

Syntax

`ReplaceTimeCodeByLocalTime = `*`Input_Value`*

`Return_Value`` = ReplaceTimeCodeByLocalTime [ `*`default: 1`*` ]`

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets replace time code by local time
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Disable</td></tr>
<tr><td>1</td><td>Enable</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns the replace time code by local time

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

objStreaming.ReplaceTimeCodeByLocalTime = 1
```

See Also

Resolution

Description

`[O]` Gets the resolution value of video configuration

Syntax

```
Return_Value = Resolution [ default: 0 ]
```

Parameters

NONE

Returns

Name	*Type*	*Description*
`Return_Value`	`Long`	Returns current resolution value in video configuration

Value	Description
0	NTSC 720x480
1	NTSC 352x240
2	NTSC 160x112
3	PAL 720x576
4	PAL 352x288
5	PAL 176x144
6	NTSC 176x120
64	NTSC 640x480
65	NTSC 1280x720
66	NTSC 1280x960
67	NTSC 1280x1024
68	NTSC 1600x1200
69	NTSC 1920x1080
70	NTSC 320x240
71	NTSC 160x120
72	NTSC 2032x1920
73	NTSC 1280x352
74	NTSC 1920x1072
192	PAL 640x480
198	PAL 320x240
199	PAL 160x112

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}

dwResolution = objStreaming.Resolution
```

See Also

`Fps, BitRate`

RTSPPort

Description

`[I/O]` Sets/Gets the special RTSP port

Syntax

```
RTSPPort = Input_Value
Return_Value = RTSPPort
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special RTSP port

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;
ObjStreaming.RTSPPort = 554 ;

objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

`RegisterPort, StreamingPort, MulticastPort, ControlPort`

Saturation

Description

`[I/O]` Sets/Gets the saturation value of video configuration

Syntax

```
Saturation = Input_Value
Return_Value = Saturation [ default: 50 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	Long	Sets the saturation value of the video configuration Minimum: 0 / Maximum: 100

Minimum	Maximum
0	100

Returns

Name	*Type*	*Description*
Return_Value	Long	Returns current saturation value in video configuration

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

```
objStreaming.Saturation = 50
dwsaturation = objStreaming.Saturation
```

See Also

Hue , Brightness

StreamingPort

Description

[I/O] Sets/Gets the streaming port

Syntax

```
StreamingPort = Input_Value
Return_Value = StreamingPort
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special streaming port

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;

objStreaming.Connect(0) ;
objStreaming.StartStream();
......
```

See Also

ControlPort, MulticastPort, HttpPort, MulticastPortRegisterPort, StreamingPort

StreamType

Description

[I/O] Sets/Gets the streaming type

Syntax

```
StreamType = Input_Value
Return_Value = StreamType
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Set the special streaming type

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
......
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.RegisterPort = 6000 ;
ObjStreaming.ControlPort = 6001 ;
ObjStreaming.StreamingPort = 6002 ;
ObjStreaming.StreamingType = 1 ;
ObjStreaming.MulticastPort = 5000 ;
ObjStreaming.HttpPort = 80 ;

objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

ControlPort, MulticastPort, HttpPort, MulticastPortRegisterPort, StreamingPort

strLastRecFileSize

Description

[o] Gets the last record file size in string

Syntax

"Return_Value" = strLastRecFileSize

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Returns the last record file size

Remarks

NONE

Example

```
ObjStreaming.strLastRecFileSize;
```

See Also

MediaUsername, Connect()

VariableFPS

Description

[I/O] Sets/Gets the variable frame per second value of video configuration

Syntax

```
VariableFPS = Input_Value
Return_Value = VariableFPS
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the variable frame per second value of the video configuration

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current variable frame per second value in video configuration

Remarks

If Fps is set to 0, then the video will not be displayed.

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
if (objStreaming.FpsMode == 1) {
     objStreaming. VariableFPS = 30 ;
}
```

See Also

BitRate, Resolution

Version

Description

[O] Gets the SDK version

Syntax

Return_Value = **Version**

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	**BSTR**	Returns the SDK version

Remarks

NONE

Example

```
// sample by javascript
alert("The current version is :" + objStreaming.Version );
```

See Also

Volume

Description

[I/O] Sets/Gets the audio volume of the PC sound device.

Syntax

```
Volume = Input_Value
Return_Value = Volume
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>Input_Value</i></td><td>LONG</td><td>Sets the audio volume of the PC sound device.
<table>
<tr><th>Minimum</th><th>Maximum</th></tr>
<tr><td>0</td><td>100</td></tr>
</table>
</td></tr>
</table>

Returns

Name	Type	Description
Return_Value	LONG	Gets the audio volume of the PC sound device.

Remarks

NONE

Example

```
......

objStreaming.Play() ;
objStreaming.Mute = 1 ;
objStreaming.Volume = 30 ;
```

See Also

Mute

DeviceType

Description

`[I/O]` Sets/Gets the device type value.

Syntax

```
DeviceType = Input_Value
Return_Value = DeviceType [ default: _SINGLE_CHANNEL_VIDEO_SERVER ]
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Input_Value</i></td><td>DeviceType</td><td>Sets the device type.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>_SINGLE_CHANNEL_VIDEO_SERVER</td></tr>
<tr><td>1</td><td>_ACD2000Q_VIDEO_SERVER</td></tr>
<tr><td>2</td><td>_SED2300Q_VIDEO_SERVER</td></tr>
<tr><td>3</td><td>_AUTO_DETECT</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	DeviceType	Returns current device type.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.DeviceType = _ACD2000Q_VIDEO_SERVER;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

QuadDeviceMode

Description

`[I/O]` Sets/Gets the device mode value on quad device.

Syntax

```
QuadDeviceMode = Input_Value
Return_Value = QuadDeviceMode [ default: SINGLE_MODE ]
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>Input_Value</i></td><td>QuadDeviceMode</td><td>Sets the device mode value on quad device.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>QUAD_MODE</td></tr>
<tr><td>1</td><td>SINGLE_MODE</td></tr>
<tr><td>2</td><td>SEQUENTIAL_MODE</td></tr>
<tr><td>3</td><td>AUTO_DETECT</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	QuadDeviceMode	Returns current device mode.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.QuadDeviceMode = QUAD_MODE;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

TCPVideoStreamID

Description

[I/O] Sets/Gets the device TCPVideoStreamID value. (Only for TCP 2.0)

Syntax

```
TCPVideoStreamID = Input_Value
Return_Value = TCPVideoStreamID [ default: 0 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	ULONG	Sets TCPVideoStreamID value. 0 ~ 255 for 1~ 256 video track

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Returns current TCPVideoStreamID.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.MediaChannel = 1;
ObjStreaming.TCPVideoStreamID = 0;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

RTPVideoTrackNumber

Description

`[I/O]` Sets/Gets the RTPVideoTrackNumber value. (Only for RTP)

Syntax

RTPVideoTrackNumber = *Input_Value*
Return_Value = RTPVideoTrackNumber [*default: 0*]

Parameters

Name	*Type*	*Description*
Input_Value	ULONG	Sets the RTPVideo track number value. 0 : Will be use 1st video track. 1~255 : Specify video track.

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Returns current setting value.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 7;    //RTP
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.MediaChannel = 1;
ObjStreaming.RTPVideoTrackNumber = 0;
ObjStreaming.RTPAudioTrackNumber = 0;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

RTPAudioTrackNumber

Description

`[I/O]` Sets/Gets the RTPAudioTrackNumber value. (Only for RTP)

Syntax

```
RTPAudioTrackNumber = Input_Value
Return_Value = RTPAudioTrackNumber [ default: 0 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	ULONG	Sets the RTPAudio track number value. 0 : Will be use 1st audio track. 1~255 : Specify audio track.

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Returns current setting value.

Remarks

NONE

Example

```
ObjStreaming.MediaType = 7;    //RTP
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
ObjStreaming.MediaChannel = 1;
ObjStreaming.RTPVideoTrackNumber = 0;
ObjStreaming.RTPAudioTrackNumber = 0;
objStreaming.Connect(0) ;
objStreaming.StartStream();
```

See Also

ConnectTimeout

Description

`[I/O]` Sets/Gets the connect timeout value.

Syntax

```
ConnectTimeout = Input_Value
Return_Value = ConnectTimeout [ default: 5 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the connect timeout value.(seconds)

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current value.(seconds)

Remarks

NONE

UI Property

Name	*Description*
Caption	`[I/O]` Sets/Gets the caption of the window
StretchToFit	`[I/O] Stretch the video window to display window`

Caption

Description

`[I/O]` Sets/Gets the caption of the window

Syntax

```
Caption = “Input_Value”
“Return_Value” = Caption
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the caption of the media window

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Returns current caption of the media window

Remarks

Notes: In preview mode, the caption always control and display by timer, about detail you can refer Desc property

Example

```
ObjStreaming.MediaType = 1;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.Caption = “Lobby” ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}
```

See Also

Play()

StretchToFit

Description

`[I/O]` Sets/Gets the display window activate or de-activate

Syntax

```
StretchToFit = Input_Value
Return_Value = StretchToFit
```

Parameters

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>Input_Value</em></td><td>LONG</td><td>Activate or de-activate current media window
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><strong>0</strong></td><td>Do not Stretch to display window</td></tr>
<tr><td><strong>1</strong></td><td>Stretch to display window</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current media window is activate or de-activate.

Remarks

NONE

Example

```
objStreaming.StretchToFit = 1 ;
```

See Also

PTZ Property

Name	***Description***
AddressID	**[I/O]** Sets/Gets current PTZ device's ID
BaudRate	**[I/O]** Sets/Gets PTZ baud rate
PTZPanSpeed	**[I/O]** Sets/Gets The PTZ pan speed
Parity	**[I/O]** Sets/Gets PTZ parity
Protocol	**[I/O]** Sets/Gets PTZ protocol
PTZFile	**[I/O]** Sets/Gets PTZ protocol file from local drive
PTZStatus	**[O]** Gets PTZ status with Enable, Disable or MousePTZ
PTZTiltSpeed	**[I/O]** Sets/Gets The PTZ tilt speed
Vendor	**[I/O]** Sets/Gets PTZ vendor

AddressID

Description

`[I/O]` Sets/Gets current PTZ device's ID

Syntax

```
AddressID = Input_Value
Return_Value = AddressID
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the PTZ device DIP switch ID

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current PTZ ID setting

Remarks

NONE

Example

```
objStreaming.MediaType = 4;
......
objStreaming.Play() ;

objStreaming.PTZFile= "C:\MyPTZ.ptz" ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = "N81" ;
objStreaming.BaudRate = 4800 ;
objStreaming.EnablePTZ() ;
```

See Also

`BaudRate, Parity`

BaudRate

Description

`[I/O]` Sets/Gets PTZ Baud rate.

Syntax

```
BaudRate = Input_Value
Return_Value = BaudRate
```

Parameters

Name	*Type*	*Description*
`Input_Value`	`LONG`	Sets/Gets PTZ Baudrate.

Value	Description
1200	Set baud rate to 1200
2400	Set baud rate to 2400
4800	Set baud rate to 4800
9600	Set baud rate to 9600
19200	Set baud rate to 19200
38400	Set baud rate to 38400
57600	Set baud rate to 57600
115200	Set baud rate to 115200
230400	Set baud rate to 230400

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	Returns current settings value

Remarks

NONE

Example

```
objStreaming.MediaType = 4;
......
objStreaming.Play() ;
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;
objStreaming.EnablePTZ() ;
```

See Also

`AddressID, Parity`

PTZPanSpeed

Description

`[I/O]` Sets/Gets PTZ pan speed

Syntax

```
PTZPanSpeed = Input_Value
Return_Value = PTZPanSpeed
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets PTZ pan speed. We can set 1~5 to change pan speed.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
ObjStreaming.MediaType = 4;
......
objStreaming.Play() ;
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;
objStreaming.PTZPanSpeed = 3;  // Set pan speed

objStreaming.EnablePTZ() ;
```

Parity

Description

`[I/O]` Sets/Gets PTZ parity

Syntax

```
Parity = “Input_Value”
“Return_Value” = Parity
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>Input_Value</i></td><td>BSTR</td><td>Sets/Gets PTZ parity.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>N81</b></td><td>No parity</td></tr>
<tr><td><b>O81</b></td><td>Set parity to odd</td></tr>
<tr><td><b>E81</b></td><td>Set parity to even</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Returns current settings value

Remarks

NONE

Example

```
ObjStreaming.MediaType = 4;
......
objStreaming.Play() ;

objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;
objStreaming.EnablePTZ() ;
```

See Also

`AddressID, BaudRate`

Protocol

Description

`[I/O]` Sets/Gets PTZ protocol.

Syntax

```
Protocol = “Input_Value”
“Return_Value” = Protocol
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the ptz protocol

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the ptz protocol

Remarks

NONE

Example

```
ObjStreaming.MediaType = 4;
......
objStreaming.Play() ;

objStreaming.Vendor = “NET-6100” ;
objStreaming.Protocol = “Pelco-P” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;
objStreaming.EnablePTZ() ;
```

See Also

`Vendor`

PTZFile

Description

`[I/O]` Sets/Gets PTZ protocol file from local drive

Syntax

```
PTZFile = “Input_Value”
“Return_Value” = PTZFile
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the ptz protocol filename form you

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the ptz protocol filename

Remarks

NONE

Example

```
objStreaming.MediaType = 4;
......
objStreaming.Play() ;

objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
```

See Also

PTZStatus

Description

[O] Gets current status of PTZ control

Syntax

Return_Value = **PTZStatus** [*default: 0*]

Parameters

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value Value / Description 0 Disable 1 Enable 2 MousePTZ 3 Absolute position

Value	Description
0	Disable
1	Enable
2	MousePTZ
3	Absolute position

Remarks

NONE

Example

```
ObjStreaming.MediaType = 4;
......
objStreaming.Play() ;

objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;

nPTZStatus = objStreaming.PTZStatus ;
```

See Also

PTZTiltSpeed

Description

[I/O] Sets/Gets PTZ tilt speed

Syntax

```
PTZTiltSpeed = Input_Value
Return_Value = PTZTiltSpeed
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets PTZ tilt speed. We can set 1~5 to change pan speed.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns current settings value

Remarks

NONE

Example

```
ObjStreaming.MediaType = 4;
......
objStreaming.Play() ;
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;
objStreaming.PTZTiltSpeed = 3;  // Set Tilt speed

objStreaming.EnablePTZ() ;
```

Vendor

Description

`[I/O]` Sets/Gets PTZ vendor

Syntax

```
Vendor = “Input_Value”
“Return_Value” = Vendor
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets the ptz protocol verdor

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets the ptz protocol verdor

Remarks

NONE

Example

```
ObjStreaming.MediaType = 4;
......
objStreaming.Play() ;

objStreaming.Vendor = “NET-6100” ;
objStreaming.Protocol = “Pelco-P” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
```

See Also

`Protocol`

Motion Detection Property

Name	*Description*
MotionDetectionAlertDuration	`[I/O]` Sets/Gets the Duration of current MD event to show red block

MotionDetectionAlertDuration

Description

`[I/O]` Sets/Gets the duration of current motion detection event to show red block

Syntax

```
MotionDetectionAlertDuration = Input_Value
Return_Value = MotionDetectionAlertDuration [ default: 10 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets the Duration of current MD event to show red block

Returns

Name	*Type*	*Description*
Return_Value	LONG	Gets the Duration of current MD event to show red block

Remarks

NONE.

Example

```
ObjStreaming.MediaType = 1 ;
......
objStreaming.Play() ;

objStreaming. MotionDetectionAlertDuration= 1 ;
objStreaming.EnableMD() ;
```

See Also

`EnableMotionDetection(), OnMDEventStart(), OnMDEventEnd(),`

DIO Property

Name	*Description*
DigitalInputInterval	**`[I/O]`** Sets/Gets the minimum interval time to issues between each DI event

DigitalInputInterval

Description

`[I/O]` Sets/Gets the minimum interval time to issues between each DI events

Syntax

```
DigitalInputInterval = Input_Value
Return_Value = DigitalInputInterval [ default: 10 ]
```

Parameters

Name	*Type*	*Description*
`Input_Value`	`LONG`	Sets the interval time between DI events

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	Gets current interval time between each DI events

Remarks

When this value is set, no digital input event will be processed within this period of time.

Example

```
ObjStreaming.MediaType = 1 ;
......
objStreaming.Play() ;

objStreaming. DigitalInputInterval= 10 ;
objStreaming.EnableDI() ;
```

See Also

`OnDIEvent()`

3 Control Methods

This section lists media methods.

Media Methods

Name	*Description*
StartAudioTransfer()	Start sending audio data to streaming server
StopAudioTransfer()	Stop sending audio data to streaming server
Connect()	Connects to the media source
DecodeFrame()	Decode Frame
DecodeFrameEx()	Decode Frame
Disconnect()	Disconnects from current media source
EnableMoudeMoveEvent()	Enable OnMouseMoveNotify() event.
FreeAudioToken()	Free the audio token.
GetAudioToken()	Get the audio token
GetBeginTimeString()	Get the begin time string of video.
GetEndTimeString()	Get the end time string of video.
Pause()	Pauses or resumes current action to the media stream or media file
Play()	Starts play current media stream or media file
PlayByFrame()	That can be used at Pause() frame by frame
SnapShot()	Saves current SnapShot image of current video frame by identity filename
SendAudio()	Send audio data to video server
SendURLCmdToSE()	Send URL command
SetCurrentTime()	Seek playback frame by time
SetDecodeI()	Set the decoder to decode I-frame only or to decode all frames
SetPlayDirection()	Set the direction of playback forward or backward
SetPlayRate()	Set the playback speed
StartAlarmRecord()	Starts recording current video to specified file name
StartDecodeMode()	Start decode mode
StartRecord()	Start play current media stream or media file
StartStream()	Starts Decoding mpeg4 stream but don’t show in the window
Stop()	Stops play current media stream or media file

StopAlarmRecord()	Stops recording current video to specified file name
StopDecodeMode	Stop decode mode.
StopRecord()	Stops recording on the current recording operation
StopPlayingAudioFile()	Stops play audio file to camera
StopStream()	Stops media streaming

StartAudioTransfer()

Description

Start sending audio data to streaming server

Syntax

```
LONG StartAudioTransfer()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Retuen_Value</i></td><td>LONG</td><td>Return setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Fail</td></tr>
<tr><td><b>1</b></td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

Please call `GetAudioControlToken()` in order to transmit audio from PC to the device

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

if (objStreaming.GetAudioToken()) {
     objStreaming.StartAudioTransfer() ;
}
```

See Also

`StopAudioTransfer()`

StopAudioTransfer()

Description

Stop sending audio data to streaming server

Syntax

```
StopAudioTransfer()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

if (objStreaming.GetAudioToken()) {
     objStreaming.StartAudioTransfer() ;
}
......

//
objStreaming. StopAudioTransfer () ;
```

See Also

```
StartAudioTransfer()
```

Connect()

Description

Connects to the media source

Syntax

```
LONG Connect( LONG AsyncConnection )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>AsyncConnection</i></td><td>LONG</td><td>Sync/Async Connection setting
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Sync</td></tr>
<tr><td>1</td><td>Async</td></tr>
</table>
</td></tr>
</table>

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Retuen_Value</i></td><td>LONG</td><td>Return connection setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.Connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

`Disconnect(), MediaPassword, MediaSource, MediaUsername, MulticastIP, Play()`

DecodeFrame()

Description

Decode MP4 data and show it on control.

Syntax

```
LONG DecodeFrame( BYTE* pFrameData, LONG nDataLen, LONG nDataType )
```

Parameters

Name	*Type*	*Description*
pFrameData	BYTE*	The frame data of video.
nDataLen	LONG	The data length of video.
nDataType	LONG	The data type of video.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......
ObjStreaming.StartDecodeMode() ;
......
ObjStreaming.DecodeFrame( pRawBuffer, nBufferLength, nDataType ) ;
......
ObjStreaming.StopDecodeMode() ;
```

See Also

DecodeFrameEx()

Description

Decode MP4 data and show it on control.

Syntax

`LONG DecodeFrameEx(OLE_HANDLE Int32BufferPointer, LONG nDataLen, LONG nDataType)`

Parameters

Name	*Type*	*Description*
Int32BufferPointer	OLE_HANDLE	The frame data of video.
nDataLen	LONG	The data length of video.
nDataType	LONG	The data type of video.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Fail</td></tr>
<tr><td><b>1</b></td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......
ObjStreaming.StartDecodeMode() ;
......
ObjStreaming.DecodeFrame( pRawBuffer, nBufferLength, nDataType ) ;
......
ObjStreaming.StopDecodeMode() ;
......
```

See Also

Disconnect()

Description

Disconnects from current media source

Syntax

```
Disconnect()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......

if (ObjStreaming.NetworkStatus >=2) {
    ObjStreaming.Disconnect() ;
} else if (ObjStreaming.NetworkStatus ==1) {
    ObjStreaming.DisConnControlOnly() ;
}
```

See Also

`Connect()`

EnableMoudeMoveEvent()

Description

Enable OnMouseMoveNotify() event.

Syntax

`EnableMoudeMoveEvent( LONG nEnable )`

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>nEnable</i></td><td>Long</td><td>Enable/Disable OnMouseMoveNotify() event.
Default is 0.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Disable</td></tr><tr><td>1</td><td>Enable</td></tr></table></td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
ObjStreaming.NetworkStatus
```

See Also

`Connect()`

FreeAudioToken()

Description

Free audio token from device.

Syntax

```
FreeAudioToken()
```

Parameters

NONE

Returns

NONE

Remarks

Please call `FreeAudioToken()` after finish audio transfer or other connection can not get the token.

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

if (objStreaming.GetAudioToken()) {
     objStreaming.StartAudioTransfer() ;
}
......
objStreaming.StopAudioTransfer() ;
objStreaming.FreeAudioToken() ;
```

See Also

GetAudioToken()

Description

Get audio token from device, only one connection can get token.

Syntax

```
LONG GetAudioToken()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Retuen_Value</i></td><td>LONG</td><td>Return setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Fail</td></tr>
<tr><td><b>1</b></td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

Please call `StartAudioTransfer()` in order to transmit audio from PC to the device

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

if (objStreaming.GetAudioToken()) {
     objStreaming.StartAudioTransfer() ;
}
```

See Also

GetBeginTimeString()

Description

Get the begin time string of video.

Syntax

```
BSTR GetBeginTimeString()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Return current begin time string of video

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
     BeginTime = objStreaming.GetBeginTimeString() ;
     EndTime = objStreaming.GetEndTimeString() ;
}
```

See Also

`GetEndTimeString()`

GetEndTimeString()

Description

Get the End time string of video.

Syntax

```
BSTR GetEndTimeString()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Return current End time string of video

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
     BeginTime = objStreaming.GetBeginTimeString() ;
     EndTime = objStreaming.GetEndTimeString() ;
}
```

See Also

`GetBeginTimeString()`

Pause()

Description

Pauses or resumes current action to the media stream or media file

Syntax

```
Pause()
```

Parameters

NONE

Returns

This function is only valid for playback mode.

Remarks

NONE

Example

```
objStreaming.MediaType = 2 ;   ‘Playback mode
objStreaming.MediaSource = “C:\abc.raw” ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

objStreaming.Pause()
```

See Also

`Play(), MediaType`

Play()

Description

Starts play current media stream or media file

Syntax

```
Play()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.Play() ;
}
```

See Also

`Stop(), MediaSource, MulticastIP, Connect()`

Note

When you call Play method, ActiveXContorl will not call Connect and StartStream automatically. AP should call Connect and StartStream before call Play method.

PlayByFrame()

Description

Starts play current media stream or media file Step by one frame

Syntax

```
PlaybyFrame( SHORT nDirection )
```

Parameters

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>nDirection</em></td><td>SHORT</td><td>Specifies the flag to play frame
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><strong>0</strong></td><td>Play backward</td></tr>
<tr><td><strong>1</strong></td><td>Play forward</td></tr>
</table>
</td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

objStreaming.Pause();

objStreaming.PlayByFrame(1); // Forward one frame
objStreaming.PlayByFrame(0); // Backward one frame
```

See Also

`Stop(), MediaSource, MulticastIP, Connect()`

SnapShot()

Description

Saves current SnapShot image of current video frame by identity filename

Syntax

```
LONG SnapShot( LONG IFormat,
               BSTR FileName,
               LONG ReduplicateTitleBarText,
               LONG R_TextColor,
               LONG G_TextColor,
               LONG B_TextColor )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>IFormat</i></td><td><b>LONG</b></td><td>Specifies the file type to SnapShot the video clips
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Bitmap format</td></tr>
<tr><td><b>1</b></td><td>JPEG format</td></tr>
<tr><td><b>2</b></td><td>GIF format</td></tr>
<tr><td><b>3</b></td><td>TIFF format</td></tr>
<tr><td><b>4</b></td><td>PNG format</td></tr>
</table></td></tr>
<tr><td><i>FileName</i></td><td><b>BSTR</b></td><td>Specifies the filename to SnapShot the video clips.
If filename is null string then will be copy bitmap to clipboard.</td></tr>
<tr><td><i>ReduplicateTitleBartext</i></td><td><b>LONG</b></td><td>Specifies the word to the picture reduplicate.</td></tr>
<tr><td><i>R_TextColor</i></td><td><b>LONG</b></td><td>Specifies the red color to the picture adjust.</td></tr>
<tr><td><i>G_TextColor</i></td><td><b>LONG</b></td><td>Specifies the green color to the picture adjust.</td></tr>
<tr><td><i>B_TextColor</i></td><td><b>LONG</b></td><td>Specifies the blue color to the picture adjust.</td></tr>
</table>

Returns

NONE

Remarks

The function now supports bmp, jpg type of formats.

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.SnapShot(1, "c:\RecTest\SnapShot" & nFile & ".jpg", 1, r_Color,
g_color, b_color) ;
}
```

See Also

`OnMDSettingChange()`

SendAudio()

Description

Saves current SnapShot image of current video frame by identity filename

Syntax

`LONG SendAudio ( BYTE* *pbyAudioBuffer*, LONG *nLength* )`

Parameters

Name	*Type*	*Description*
pbyAudioBuffer	BYTE*	Audio data that need to send
nLength	LONG	Length of audio buffer

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

The audio data will be sent by control port.

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     // Send audio data to video server
     objStreaming.SendAudio(pbyAudioBuff, lLen)
}
```

See Also

SendURLCmdToSE()

Description

Send URL command to video server.

Syntax

`BSTR SendURLCmdToSE ( BSTR szCommand )`

Parameters

Name	*Type*	*Description*
szCommand	BSTR	URL command will be sent

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Return setting result.

Remarks

NONE

Example

```
// Send to video server, that will get result string from pbstrReturn
ObjStreaming.SendURLCmdToSE("http\\192.168.1.100\");
```

See Also

SetCurrentTime()

Description

Sets current time in playback mode

Syntax

```
SetCurrentTime ( LONG nCurrentTime )
```

Parameters

Name	*Type*	*Description*
nCurrentTime	LONG	Seek position to set time

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.Play();
objStreaming. SetCurrentTime (nCurrentTime) ;
```

See Also

SetDecodeI()

Description

Set the decoder to decode I frame only or to decode all frames

Syntax

```
SetDecodeI( SHORT DecodeIOnly )
```

Parameters

Name	*Type*	*Description*
DecodeIOnly	SHORT	Specifies the flag to decode frames **Value** – **Description** **0** – All frames (default value) **1** – I frame only

Returns

NONE

Remarks

When application calls this function, Media Control will decode I-Frame only (without decoding P-Frame), in this way, the CPU loading will decrease dramatically

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.SetDecodeI(1) ;
     objStreaming.Play();
}
```

See Also

`Play()`

SetPlayDirection()

Description

Sets current play direction in the playback media

Syntax

```
SetPlayDirection( SHORT shDirection )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>shDirection</i></td><td>SHORT</td><td>Specifies the flag to decode frames<table><tr><th>Value</th><th>Description</th></tr><tr><td><b>0</b></td><td>Play Backward</td></tr><tr><td><b>1</b></td><td>Play Forward.</td></tr></table></td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......

objStreaming.Play() ;
objStreaming.SetPlayDirection(1) ; // Play Forward

objStreaming.Pause() ;
objStreaming.SetPlayDirection(0) ; // Play Backward
objStreaming.Play() ;
```

See Also

SetPlayRate()

Description

Set playback speed

Syntax

```
SetPlayRate( LONG nPlayRate )
```

Parameters

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>nPlayRate</em></td><td>LONG</td><td>Set speed of playback file
<table>
<tr><th>Value</th><th>Speed</th></tr>
<tr><td>0</td><td>1/2x</td></tr>
<tr><td>1</td><td>1x</td></tr>
<tr><td>2</td><td>2x</td></tr>
<tr><td>3</td><td>4x</td></tr>
<tr><td>4</td><td>8x</td></tr>
</table>
</td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......

objStreaming.Play() ;
objStreaming.SetPlayRate(2) ; // Set playback speed to 2x
```

See Also

StartAlarmRecord()

Description

Enable the alarm recording job

Syntax

```
LONG StartAlarmRecord( BSTR szFileName )
```

Parameters

Name	*Type*	*Description*
szFileName	BSTR	Specifies the filename to record the video clips

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.StartAlarmRecord(“C:\ABC.raw”);
}
```

See Also

`StartRecord()`, `OnStopRecord()`, `StopAlarmRecord()`, `PreRecordTime`, `PostRecordingTime`

StartDecodeMode ()

Description

Starts decoding current video to media mode.

Syntax

```
LONG StartDecodeMode()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
ObjStreaming.StartDecodeMode() ;
......
ObjStreaming.DecodeFrame( pRawBuffer, nBufferLength, nDataType ) ;
......
ObjStreaming.StopDecodeMode() ;
```

See Also

StartRecord()

Description

Starts recording current video to specified file name

Syntax

```
LONG StartRecord( BSTR szFileName )
```

Parameters

Name	*Type*	*Description*
szFileName	BSTR	Specifies the filename to record the video clips

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result. Value / Description 0 / Fail 1 / Success

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
BOOL bRecording = false;
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.StartRecord(“C:\ABC.raw”);
     bRecording = true;
}
If(bRecording){
     objStreaming.StopRecord();
}
```

See Also

`StopRecord()`

StartStream()

Description

Starts Decoding mpeg4 stream but don’t show in the window

Syntax

```
StartStream()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
ObjStreaming.ID = 1;
ObjStreaming.MediaSource = “192.168.0.100” ;
ObjStreaming.MediaUsername = “Admin” ;
ObjStreaming.MediaPassword = “123456” ;
objStreaming.AutoReconnect = 1 ;
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.StartStream();
}
```

See Also

```
StopRecord()
```

Stop()

Description

Stops play current media stream or media file

Syntax

`Stop()`

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.connect(0) ;
objStreaming.StartStream();

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.Play() ;
}

......

If (ObjStreaming.ContentStatus == 1) {
     objStreaming.Stop() ;
}
```

See Also

`Play()`

StopAlarmRecord()

Description

Stops alarm recording job

Syntax

```
StopAlarmRecord()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......

if (objStreaming.RecordType == 2) {
     objStreaming.StopAlarmRecord() ;
} else if (objStreaming.RecordType == 1) {
     objStreaming.StopRecord() ;
}
```

See Also

`StartRecord()`, `OnStopRecord()`, `StartAlarmRecord()`, `PreRecordTime`, `PostRecordingTime`

StopDecodeMode()

Description

Stops decoding current video to media mode

Syntax

```
StopDecodeMode()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
ObjStreaming.StartDecodeMode() ;
......
ObjStreaming.DecodeFrame( pRawBuffer, nBufferLength, nDataType ) ;
......
ObjStreaming.StopDecodeMode() ;
```

See Also

```
StartDecodeMode ()
```

StopRecord()

Description

Stops recording on the current recording operation

Syntax

```
StopRecord()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......

if (objStreaming.RecordType == 2) {
     objStreaming.StopAlarmRecord() ;
} else if (objStreaming.RecordType == 1) {
     objStreaming.StopRecord() ;
}
```

See Also

`StartRecord(), OnStopRecord()`

StopPlayingAudioFile()

Description

Stop playing audio file to camera.

Syntax

```
StopPlayingAudioFile()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.MediaType = 6
......
objStreaming.PlayAudioFileToDevice(“C:\Audio.wav”)
......
objStreaming.StopPlayingAudioFile()
```

See Also

`PlayAudioFileToDevice()`

StopStream()

Description

Stop Decoding mpeg4 stream

Syntax

```
StopStream()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.connect(0) ;

If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.StartStream() ;
}

......

If (ObjStreaming.ContentStatus == 1) {
     objStreaming.StopStream() ;
}
```

See Also

```
StopRecord()
```

SetVideoTransformConfig()

Description

Set the video transform config to force streaming engine resize video.

Syntax

```
SetVideoTransFormConfig( USHORT bTransfer,
                         LONG nVideoOutType,
                         LONG BitRate,
                         LONG nFpsNum,
                         USHORT bReSize,
                         LONG nResolution )
```

Parameters

Name	*Type*	*Description*
bTransfer	USHORT	0 is false 1 is true
nVideoOutType	LONG	1 : Mpeg4 4 : MJPEG 5 : H.264
BitRate	LONG	Ex:500000 is 500kbps
nFpsNum	LONG	Frame rate
bReSize	USHORT	0 is disable, 1 is enable.
nResolution	LONG	Refer to Resolution map

Returns

NONE

Remarks

See Also

UI Methods

Name	*Description*
DisplayTitleBar()	Show or hide title bar of the media window
EnableFullScreen()	Enable the FullScreen
SetControlActive()	Activate or de-activate current media window
SetDefaultBackgroundColor ()	Set default background color
SetTitleBarTextLayout()	Set the title bar text layout
SetTitleActiveColor()	Set the title color in active mode.
SetTitleNonActiveColor()	Set the title color in Non-active mode.
SetTitleEventColor()	Set the title color in event mode
SetPlayingBackgroundColor()	Set the Playing background color
SetBorderWidth	Set the control border width.
SetBorderColor	Set the control border color.

DisplayTitleBar()

Description

Show or hide title bar of the media window

Syntax

```
DisplayTitleBar( LONG Display )
```

Parameters

Name	Type	Description
Display	LONG	Show or hide title bar of the media window Value / Description: **0** – Hide the title bar of the media window **1** – Display the title bar of the media window

Value	Description
0	Hide the title bar of the media window
1	Display the title bar of the media window

Returns

NONE

Remarks

⚠ **NOTE:** This function is useful when you want to develop a multiple-channel preview or playback, e.g. 16-channel preview, and the preview window is too small to accommodate any title bar

Example

```
'---- display title bar
objStreaming.DisplayTitleBar(1) ;
```

See Also

EnableFullScreen()

Description

Enable the FullScreen.

Syntax

```
EnableFullScreen()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
objStreaming.EnableFullScreen () ;
```

See Also

SetControlActive()

Description

Activate or de-activate current media window

Syntax

```
SetControlActive( LONG nActive )
```

Parameters

Name	*Type*	*Description*
nActive	LONG	Activate or de-activate current media window Value / Description 0 — Set the window to de-active 1 — Set the window to Active

Value	Description
0	Set the window to de-active
1	Set the window to Active

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetControlActive (1) ;
```

See Also

SetDefaultBackgroundColor ()

Description

Set the default background color.

Syntax

```
SetDefaultBackgroundColor( LONG R, LONG G, LONG B )
```

Parameters

Name	*Type*	*Description*
R	LONG	0~255
G	LONG	0~255
B	LONG	0~255

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetDefaultBackgroundColor (255, 255, 255) ;
```

See Also

SetTitleBarTextLayout()

Description

Set the title bar text layout

Syntax

```
SetTitleBarTextLayout( LONG n1,
                       LONG n2,
                       LONG n3,
                       LONG n4,
                       LONG n5,
                       LONG n6,
                       LONG n7,
                       LONG n8,
                       LONG n9,
                       LONG n10,
                       LONG n11,
                       LONG n12,
                       LONG n13,
                       LONG n14,
                       LONG n15,
                       LONG n16,
                       LONG n17,
                       LONG n18,
                       LONG n19,
                       LONG n20 );
```

Parameters

Name	*Type*	*Description*
n1	LONG	
n2	LONG	
n3	LONG	
n4	LONG	
n5	LONG	
n6	LONG	
n7	LONG	
n8	LONG	
n9	LONG	
n10	LONG	

n11	**LONG**
n12	**LONG**
n13	**LONG**
n14	**LONG**
n15	**LONG**
n16	**LONG**
n17	**LONG**
n18	**LONG**
n19	**LONG**
n20	**LONG**

Value	Description
0	NONE
1	YYYY EX:2008
2	YY EX:08
3	MM EX:12
4	DD EX:28
5	hh EX:23
6	mm EX:29
7	ss EX:59
8	weekday EX:”Mon”
9	ID EX:1
10	Caption EX:”Front Door”
11	Slash EX: “/“
12	Colon EX: “:”
13	At EX: “@”
14	Dash EX:”-”
15	LeftBracket EX: “[“
16	RightBracket EX: ”]”
17	Space EX: “ “
18	Space 3 EX: “ “
19	Space 5 Ex: “ “
20	FPS
21	ms

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetTitleBarTextLayout(9, 12, 10, 19,
1, 14, 2, 14, 3, 18, 5, 12, 6, 12, 7, 0, 0, 0, 0,
0) ;
```

See Also

SetTitleActiveColor()

Description

Set the title color in active mode. Title is consisting of 16 lines.

Syntax

```
SetTitleActiveColor( LONG nIndex, LONG R, LONG G, LONG B )
```

Parameters

Name	*Type*	*Description*
nIndex	LONG	Line number is range from 0~15.
R	LONG	0~255
G	LONG	0~255
B	LONG	0~255

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetTitleActiveColor(1, 255, 255, 255) ;
```

See Also

SetTitleNonActiveColor()

Description

Set the title color in non-active mode. Title is consisting of 16 lines

Syntax

```
SetTitleNonActiveColor( LONG nIndex, LONG R, LONG G, LONG B)
```

Parameters

Name	*Type*	*Description*
nIndex	LONG	Line number is range from 0~15.
R	LONG	0~255
G	LONG	0~255
B	LONG	0~255

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetTitleNonActiveColor(1, 255, 255, 255) ;
```

See Also

SetTitleEventColor()

Description

Set the title color in event mode. Title is consist of 16 lines

Syntax

```
SetTitleEventColor( LONG nIndex, LONG R, LONG G, LONG B)
```

Parameters

Name	*Type*	*Description*
nIndex	LONG	Line number is range from 0~15.
R	LONG	0~255
G	LONG	0~255
B	LONG	0~255

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetTitleEventColor(1, 255, 255, 255) ;
```

See Also

SetPlayingBackgroundColor()

Description

Set the playing background color

Syntax

```
SetPlayingBackgroundColor( LONG R, LONG G, LONG B )
```

Parameters

Name	*Type*	*Description*
R	LONG	0~255
G	LONG	0~255
B	LONG	0~255

Returns

NONE

Remarks

NONE

Example

```
objStreaming.SetPlayingBackgroundColor (255, 255, 255) ;
```

See Also

SetBorderWidth()

Description

Set the control border width.

Syntax

```
SetBorderWidth( LONG value )
```

Parameters

Name	*Type*	*Description*
value	LONG	Specifies the width of the pen, in logical units.

Returns

NONE

Remarks

See Also

SetBorderColor()

Description

Set the control border color.

Syntax

```
SetBorderColor( LONG colorR, LONG colorG, LONG colorB )
```

Parameters

Name	*Type*	*Description*
colorR	LONG	Specifies the intensity of the red color.
colorG	LONG	Specifies the intensity of the green color.
colorB	LONG	Specifies the intensity of the blue color.

Returns

NONE

Remarks

See Also

PTZ Methods

Name	*Description*
DigitalZoomIn()	Operate the Digital Zoom In functions
DigitalZoomOut()	Operate the Digital Zoom Out functions
DigitalMoveLeft ()	Operate the Digital Move Left functions
DigitalMoveRight ()	Operate the Digital Move Right functions
DigitalMoveUp ()	Operate the Digital Move Up functions
DigitalMoveDown ()	Operate the Digital Move Down functions
DigitalZoomOutMax ()	Operate the Digital Zoom Out Max functions
DisableMouseDigitalPTZ()	Disable Mouse Digital PTZ function
DisableMousePTZ()	Disable Mouse PTZ functions
DisablePTZ()	Disable PTZ functions
EnableMousePTZ()	Enable Mouse PTZ functions
EnableMouseDigitalPTZ()	Enable Mouse Digital PTZ function
EnablePTZ()	Enable PTZ functions
EnumerateProtocol()	Enumerate available protocol
PTZBLC()	Operate the camera BLC function
PTZDayNight()	Operate the camera day&night function
PTZFocus()	Operate the camera focus function
PTZIris()	Operate the camera iris function
PTZMove()	Move the camera view
PTZOSD()	Operate the camera OSD function
PTZPreset()	Operate the camera preset functions
PTZZoom()	Operate the camera Zoom functions
SendPTZCmd()	Send PTZ protocol command to video server or speed dome

DigitalZoomIn()

Description

Operate the Digital Zoom In functions

Syntax

```
LONG DigitalZoomIn( LONG nSteps )
```

Parameters

Name	*Type*	*Description*
nSteps	LONG	Set the special the digital zoom in value.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>Fail</td></tr><tr><td>1</td><td>Success</td></tr></table></td></tr>
</table>

Remarks

NONE

Example

```
objStreaming.DigitalZoomIn(LONG nSteps)
```

See Also

PTZZoom()

DigitalZoomOut()

Description

Operate the Digital Zoom In functions

Syntax

```
LONG DigitalZoomOut( LONG nSteps )
```

Parameters

Name	*Type*	*Description*
nSteps	LONG	Set the special the digital zoom Out value.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
objStreaming.DigitalZoomOut(LONG nSteps)
```

See Also

`PTZZoom()`

DigitalMoveLeft ()

Description

Operate the Digital Zoom In functions

Syntax

```
LONG DigitalMoveLeft( LONG nSteps )
```

Parameters

Name	*Type*	*Description*
nSteps	LONG	Set the special the digital move left value.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
objStreaming.DigitalMoveLeft(LONG nSteps);
```

See Also

`PTZZoom()`

DigitalMoveRight ()

Description

Operate the Digital Zoom In functions

Syntax

```
LONG DigitalMoveRight( LONG nSteps )
```

Parameters

Name	*Type*	*Description*
nSteps	LONG	Set the special the digital move right value.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
objStreaming.DigitalMoveRight(LONG nSteps);
```

See Also

PTZZoom()

DigitalMoveUp ()

Description

Operate the Digital Zoom In functions

Syntax

```
LONG DigitalMoveUp( LONG nSteps )
```

Parameters

Name	*Type*	*Description*
nSteps	LONG	Set the special the digital move Up value.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
objStreaming.DigitalMoveUp(LONG nSteps);
```

See Also

`PTZZoom()`

DigitalMoveDown ()

Description

Operate the Digital Zoom In functions

Syntax

```
LONG DigitalMoveDown( LONG nSteps )
```

Parameters

Name	*Type*	*Description*
nSteps	LONG	Set the special the digital move down value.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result. <table><tr><th>Value</th><th>Description</th></tr><tr><td>**0**</td><td>Fail</td></tr><tr><td>**1**</td><td>Success</td></tr></table>

Remarks

NONE

Example

```
objStreaming.DigitalMoveDown(LONG nSteps);
```

See Also

`PTZZoom()`

DigitalZoomOutMax ()

Description

Operate the Digital Zoom In functions

Syntax

```
DigitalZoomOutMax ()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
objStreaming.DigitalZoomOutMax();
```

See Also

`PTZZoom()`

DisableMouseDigitalPTZ()

Description

Disable Mouse Digital PTZ functions

Syntax

```
DisableMouseDigitalPTZ()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
objStreaming.EnableMouseDigitalPTZ() ;

objStreaming.DisableMouseDigitalPTZ() ;
objStreaming.DisablePTZ() ;
```

See Also

`EnableMouseDigitalPTZ()`

DisableMousePTZ()

Description

Disable Mouse PTZ functions

Syntax

```
DisableMousePTZ()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
objStreaming.EnableMousePTZ() ;

objStreaming.DisableMousePTZ() ;
objStreaming.DisablePTZ() ;
```

See Also

`EnableMousePTZ()`

DisablePTZ()

Description

Disable PTZ functions

Syntax

```
DisablePTZ()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......

objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;

objStreaming.DisablePTZ() ;
```

See Also

`EnablePTZ()`

EnableMousePTZ()

Description

Enable Mouse PTZ functions

Syntax

```
LONG EnableMousePTZ()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
objStreaming.EnableMousePTZ() ;

objStreaming.DisableMousePTZ() ;
objStreaming.DisablePTZ() ;
```

See Also

`DisableMousePTZ()`

EnableMouseDigitalPTZ()

Description

Enable Mouse Digital PTZ functions

Syntax

```
EnableMouseDigitalPTZ()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;
objStreaming.EnableMouseDigitalPTZ() ;

objStreaming.DisableMouseDigitalPTZ() ;
```

See Also

`DisableMouseDigitalPTZ()`

EnablePTZ()

Description

Enable PTZ functions

Syntax

```
LONG EnablePTZ()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.<table><tr><th>Value</th><th>Description</th></tr><tr><td><b>0</b></td><td>Fail</td></tr><tr><td><b>1</b></td><td>Success</td></tr></table></td></tr>
</table>

Remarks

NONE

Example

```
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;

objStreaming.DisablePTZ() ;
```

See Also

`DisablePTZ()`

EnumerateVendor

Description

Enumerate available Vendor that was changed from method to property.

Syntax

```
BSTR EnumerateVendor()
```

Remarks

NONE

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Available vendor list

Remarks

We can use the property to get vendor that was supported and it was formatted as first number is how many vendors were supported. Every vendor was separate with ‘,’. For example, 3,CAM-6100,CAM-6200,CAM-6500. First number 3 means there have 3 vendor fellows.

Example

```
strVendor = objStreaming.EnumerateVendor(); // Get available vendor

objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
```

See Also

`EnumerateProtocol()`

EnumerateProtocol()

Description

Enumerate available protocol

Syntax

```
BSTR EnumerateProtocol( BSTR szVendor )
```

Parameters

Name	*Type*	*Description*
szVendor	BSTR	Specifies the vendor

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Returns the protocol listing of the vendor specified

Remarks

NONE

Example

```
objStreaming.PTZFile= “C:\MyPTZ.ptz” ;
objStreaming.AddressID = 1 ;
objStreaming.Parity = “N81” ;
objStreaming.BaudRate = 4800 ;

objStreaming.EnablePTZ() ;
szProtocol = objStreaming.EnumerateProtocol(“NET-6100”);
```

See Also

PTZBLC()

Description

Operate the camera BLC function

Syntax

```
PTZBLC( BSTR szInputCommand )
```

Parameters

Name	*Type*	*Description*
szInputCommand	**BSTR**	Move the camera view **Value** – **Description** **ON** – BLC on **OFF** – BLC off

Value	Description
ON	BLC on
OFF	BLC off

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming. PTZBLC("ON") ;
//
objStreaming.PTZBLC("OFF") ;
```

See Also

PTZDayNight()

Description

Operate the camera Day and Night function

Syntax

```
PTZDayNight( BSTR szInputCommand )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>szInputCommand</i></td><td>BSTR</td><td>Move the camera view
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>On</b></td><td>Day and night function on</td></tr>
<tr><td><b>OFF</b></td><td>Day and night function off</td></tr>
<tr><td><b>AUTOON</b></td><td>Day and night function auto on</td></tr>
<tr><td><b>AUTOOFF</b></td><td>Day and night function auto off</td></tr>
</table></td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming. PTZDayNight(“ON”) ;
//
objStreaming.PTZDayNight(“OFF”) ;
```

See Also

PTZFocus()

Description

Operate the camera Focus function

Syntax

```
PTZFocus( BSTR szInputCommand )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>szInputCommand</i></td><td>BSTR</td><td>Move the camera view
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>IN</b></td><td>Focus In</td></tr>
<tr><td><b>OUT</b></td><td>Focus Out</td></tr>
<tr><td><b>STOP</b></td><td>Stop Focus Action</td></tr>
</table></td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming. PTZDayNight(“ON”) ;
//
objStreaming.PTZDayNight(“OFF”) ;
```

See Also

PTZIris()

Description

Operate the camera iris function

Syntax

```
PTZIris( BSTR szInputCommand )
```

Parameters

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>szInputCommand</em></td><td>BSTR</td><td>Move the camera view
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><strong>OPEN</strong></td><td>Open iris</td></tr>
<tr><td><strong>CLOSE</strong></td><td>Close iris</td></tr>
<tr><td><strong>STOP</strong></td><td>Stop iris</td></tr>
<tr><td><strong>AUTO</strong></td><td>Open auto iris</td></tr>
</table></td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming. PTZIris(“OPEN”) ;
//
objStreaming.PTZIRIS(“CLOSE”) ;
```

See Also

PTZMove()

Description

Move the camera view

Syntax

`PTZMove( BSTR szInputCommand )`

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><i>szInputCommand</i></td><td>BSTR</td><td>Move the camera view
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>LEFT</td><td>Pan Left</td></tr>
<tr><td>RIGHT</td><td>Pan Right</td></tr>
<tr><td>UP</td><td>Tilt Up</td></tr>
<tr><td>DOWN</td><td>Tilt Down</td></tr>
<tr><td>STOP</td><td>Stop moving</td></tr>
<tr><td>HOME</td><td>Back to home position</td></tr>
<tr><td>UPLEFT</td><td>Pan left and tilt up</td></tr>
<tr><td>UPRIGHT</td><td>Pan right and tilt up</td></tr>
<tr><td>DOWNLEFT</td><td>Pan left and tilt down</td></tr>
<tr><td>DOWNRIGHT</td><td>Pan right and tilt down</td></tr>
</table>
</td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming.PTZMOVE(“LEFT”) ;
//
objStreaming.PTZMOVE(“RIGHT”) ;
//
objStreaming.PTZMOVE(“UP”) ;
//
objStreaming.PTZMOVE(“DOWN”) ;
//
```

```
objStreaming.PTZMOVE(“UPRIGHT”) ;
objStreaming.PTZMOVE(“UPLEFT”) ;
objStreaming.PTZMOVE(“DOWNRIGHT”) ;
objStreaming.PTZMOVE(“DOWNLEFT”) ;
//
objStreaming.PTZMOVE(“STOP”) ;
//
objStreaming.PTZMOVE(“HOME”) ;
```

See Also

PTZOSD()

Description

Operate the camera OSD function

Syntax

```
PTZOSD( BSTR szInputCommand )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>szInputCommand</i></td><td>BSTR</td><td>Move the camera view
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>ON</b></td><td>Open OSD</td></tr>
<tr><td><b>OFF</b></td><td>Close OSD</td></tr>
<tr><td><b>LEFT</b></td><td>OSD Cursor move left</td></tr>
<tr><td><b>RIGHT</b></td><td>OSD Cursor move right</td></tr>
<tr><td><b>UP</b></td><td>OSD Cursor move up</td></tr>
<tr><td><b>DOWN</b></td><td>OSD Cursor move down</td></tr>
<tr><td><b>ENTER</b></td><td>Enter OSD sub menu</td></tr>
<tr><td><b>LEAVE</b></td><td>Leave OSD menu</td></tr>
</table></td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming. PTZOSD(“ON”) ;
//
objStreaming.PTZOSD(“OFF”) ;
```

See Also

PTZPreset()

Description

Operate the camera preset functions

Syntax

`PTZPreset( BSTR szInputCommand, LONG nPresetIndex )`

Parameters

Name	*Type*	*Description*
szInputCommand	BSTR	Move the camera view **Value** / **Description** **SET** / Set the preset position **CLEAR** / Clear the preset position **GOTO** / Goto the preset position
nPresetIndex	LONG	Move the camera view **Minimum** / **Maximum** 1 / 16

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming. PTZPreset(“SET”, 1) ;
//
objStreaming. PTZPreset(“CLEAR”, 1) ;
//
objStreaming. PTZPreset(“GOTO”, 1) ;
```

See Also

PTZZoom()

Description

Operate the camera Zoom functions

Syntax

```
PTZZoom( BSTR szInputCommand )
```

Parameters

Name	*Type*	*Description*
szInputCommand	BSTR	Move the camera view

Value	Description
IN	Zoom in
OUT	Zoom out
STOP	Stop moveing

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
objStreaming.PTZSetPanSpeed(3) ;
objStreaming.PTZSetTiltSpeed(3) ;
objStreaming.PTZZOOM(“IN”) ;
//
objStreaming.PTZZOOM(“OUT”) ;
//
objStreaming.PTZSTOP(“STOP”) ;
```

See Also

SendPTZCmd()

Description

Send ASCII string command to video server

Syntax

```
LONG SendPTZCmd( BSTR szCommand, LONG nParam1, LONG nParam2 )
```

Parameters

Name	*Type*	*Description*
szCommand	BSTR	The PTZ command that is defined by protocol file
nParam1	SHORT	The Param1 that is defined by protocol file
nParam2	SHORT	The Param2 that is defined by protoco2 file

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
<tr><td>2</td><td>Else</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
objPTZ.SendPTZCmd("PANLEFT", 3, 0); // Send a PANLEFT command, speed 3 to speed
dome.
```

See Also

Motion Detection Methods

Name	*Description*
DisableMotionDetection()	Disable or delete motion detection area
EnableMotionDetection()	Enable or add new motion detection area
GetMotionSensitive()	Get motion sensitive value.
GetMotionStartX()	Get motion start X position
GetMotionStartY()	Get motion startY position
GetMotionEndX()	Get motion end X position
GetMotionEndY()	Get motion end Y position
SetMotionSetting()	Set motion detection settings
SetQuadMotionDetection()	Set quad motion detection.
StartMDSetup()	Set the motion status to SETTING mode
StopMDSetup()	Set the motion status to normal mode
SyncMDInfoFromDevice()	Synchronize motion detect setting value on ip device

DisableMotionDetection()

Description

Disable motion detection

Syntax

```
DisableMotionDetection()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

`EnableMotionDetection()`

EnableMotionDetection()

Description

Enable motion detection

Syntax

```
LONG EnableMotionDetection()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

GetMotionSensitive()

Description

Get motion sensitive value

Syntax

```
LONG GetMotionSensitive( LONG nRegionNumber )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Set the special the region number.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;
MotionSensitive = objStreaming.GetMotionSensitive (100) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

SetMotionSensitivity()

GetMotionStartX()

Description

Get motion start X position

Syntax

```
LONG GetMotionStartX( LONG nRegionNumber )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Set the special the region number.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;
StartX = objStreaming.GetMotionStartX (100);

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

GetMotionStartY()

Description

Get motion start Y position

Syntax

```
LONG GetMotionStartY( LONG nRegionNumber )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Set the special the region number.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;
StartX = GetMotionStartY (100);

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

GetMotionEndX()

Description

Get motion end X position

Syntax

```
LONG GetMotionEndX( LONG nRegionNumber )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Set the special the region number.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;
EndX = GetMotionEndX (100);

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

GetMotionEndY()

Description

Get motion end Y position

Syntax

```
LONG GetMotionEndY( LONG nRegionNumber )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Set the special the region number.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;
EndY = GetMotionEndY (100);

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//
objStreaming.DisableMotionDetection () ;
```

See Also

GetMotionRegionEnable()

Description

Get motion sensitive value

Syntax

```
ULONG GetMotionRegionEnable( LONG nRegionNumber )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Set the special the region number.

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Returns current motion enable status.

Remarks

NONE

SetMotionRegionEnable()

Description

Set motion enable status

Syntax

```
SetMotionRegionEnable( LONG nRegionNumber, ULONG ulEnable )
```

Parameters

Name	*Type*	*Description*
nRegionNumber	LONG	Motion detection window number. Valid values are: 1, 2, 3
ulEnable	ULONG	Enable status

Returns

NONE

Remarks

NONE

SetMotionSetting()

Description

Set motion detection settings

Syntax

```
SetMotionSetting( LONG nMotionRegionNumber,
                  LONG StartX,
                  LONG StartY,
                  LONG EndX,
                  LONG EndY,
                  LONG nSesitive )
```

Parameters

Name	*Type*	*Description*
nMotionRegionNumber	LONG	Motion detection window number. Valid values are: 1, 2, 3
StartX	LONG	Motion detection window region starting X value
StartY	LONG	Motion detection window region starting Y value
EndX	LONG	Motion detection window region ending X value
EndY	LONG	Motion detection window region ending Y value
nSesitive	LONG	Motion detection window region sensitivity

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//---- motion detection callback will be passed
objStreaming.DisableMotionDetection() ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

SetQuadMotionDetection()

Description

Set Quad Motion Detection

Syntax

```
LONG SetQuadMotionDetection( BYTE byEnable )
```

Parameters

Name	*Type*	*Description*
byEnable	BYTE	Set quad motion detection number.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......
objStreaming.StartMDSetup() ;

objStreaming.SetMotionSensitivity(1,80) ;
QuadMotionDetection = objStreaming.SetQuadMotionDetection(1);

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//---- motion detection callback will be passed
objStreaming.DisableMotionDetection() ;
```

See Also

`SetMotionSensitivity()`

SetMotionSensitivity()

Description

Set motion detection settings

Syntax

`SetMotionSensitivity( LONG nMotionRegionNumber, LONG nSensitive )`

Parameters

Name	*Type*	*Description*
nMotionRegionNumber	LONG	Motion detection window number. Valid values are: 1, 2, 3
nSensitive	LONG	Motion detection window region starting X value

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//---- motion detection callback will be passed
objStreaming.DisableMotionDetection() ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

StartMDSetup()

Description

Set the motion status to setting mode

Syntax

```
LONG StartMDSetup()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//---- motion detection callback will be passed
objStreaming.DisableMotionDetection() ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

StopMDSetup()

Description

Set the motion status to normal mode

Syntax

```
StopMDStup()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......

objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//---- motion detection callback will be passed
objStreaming.DisableMotionDetection() ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

SyncMDInfoFromDevice()

Description

Synchronize motion detect setting value on ip device.

Syntax

```
LONG SyncMDInfoFromDevice()
```

Parameters

NONE

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

Synchronize settings in first time.

Example

```
......
SyncMDInfoFromDevice() ;
objStreaming.StartMDSetup() ;

objStreaming.SetMotionSetting(1,5,5,100,100,50) ;
objStreaming.SetMotionSetting(2,105,105,150,150,30) ;
objStreaming.SetMotionSensitivity(1,80) ;

objStreaming.StopMDSetup() ;

objStreaming.EnableMotionDetection () ;
//---- motion detection callback will be passed
objStreaming.DisableMotionDetection() ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

DIO Methods

Name	*Description*
DisableDigitalInput()	Enable digital input that will receive data from video server
EnableDigitalInput()	Disable digital input that will not receive data from video server
DigitalOutput	Trigger assigned digital out event

DisableDigitalInput()

Description

Disable to receive Digital input event

Syntax

```
DisableDigitalInput()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
If (ObjStreaming.NetworkStatus != 0) {
     objStreaming.StartStream();
}

objStreaming.EnableDigitalInput () ;
//
objStreaming.DisableDigitalInput () ;
```

See Also

`EnableMotionDetection()`

EnableDigitalInput()

Description

Enable and receive digital input event

Syntax

```
LONG EnableDigitalInput()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Return one.

Remarks

NONE

Example

```
......
If (ObjStreaming.NetworkStatus != 0) {
    objStreaming.StartStream();
}

objStreaming.EnableDigitalInput () ;
//
objStreaming.DisableDigitalInput () ;
```

See Also

`DisableMotionDetection(), OnMDEventStart()`

DigitalOutput

Description

Trigger assigned digital out event

Syntax

```
DigitalOutput( LONG nDO1, LONG nDO2, LONG nDO3, LONG nDO4 )
```

Parameters

Name	***Type***	***Description***
nDO1	LONG	Set to 0 will disable the Digital Output 1, Set to 1 will enable Digital Output 1 signal.
nDO2	LONG	Set to 0 will disable the Digital Output 2, Set to 1 will enable Digital Output 2 signal.
nDO3	LONG	*(Reserved)* Set to 0 will disable the Digital Output 3, Set to 1 will enable Digital Output 3 signal.
nDO4	LONG	*(Reserved)* Set to 0 will disable the Digital Output 4, Set to 1 will enable Digital Output 4 signal.

Returns

NONE

Remarks

NONE

Example

```
objDIO.DigitalOutput(1,0,0,0)
```

See Also

Other Methods

Name	*Description*

4 Control Events

This section lists media events. These values maybe triggered or customized.

UI Events

Name	*Description*
OnConnect()	Event generated when the control connect success
OnConnectionRecovery()	Event generated when the control reconnect successfully
OnDisconnect()	Event generated when the control disconnect
OnMouseKeyDoubleClick()	Event generated when the user double clicks on the media area in the media window
OnMouseKeyDown()	Event generated when the mouse down on the media area in the media window
OnMouseKeyUp()	Event generated when the mouse up on the media area in the media window
OnMouseMoveNotify()	Event generated when the mouse left button down and move on the media area in the media window
OnNetworkLoss()	Event generated when the connection break.
OnPlay()	Event generated when the control start preview
OnPlaybackComplete()	Event generated when the control playback file complete
OnSaveImage()	Event generated when save a image
OnStartRecord()	Event generated when recording starts
OnStop()	Event generated when the control stop preview
OnStopRecord()	Event generated when recording stops
OnTimeCode()	Event generated when time code was be updated
OnVideoLoss()	Event generated when video loss occurs
OnVideoRecovery()	Event generated when video recovery occurs after video loss

OnConnect()

Description

Event generated when the control connect success.

Syntax

```
OnConnect( LONG nID, LONG ConnectSuccessful )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>nID</i></td><td>Long</td><td>Return camera id</td></tr>
<tr><td><i>ConnectSuccessful</i></td><td>Long</td><td>The connect return code
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Connect failed</td></tr>
<tr><td><b>1</b></td><td>Connect successful</td></tr>
</table>
</td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT FOR=Media EVENT=”OnConnect(nID,successful)” LANGUAGE=”Jscript”>
    Alert( “Connect Success” );
</SCRIPT>
‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnConnect()
        MsgBox “Connect Success”
    End Function
</SCRIPT>
```

See Also

OnConnectionRecovery()

Description

Event generated when the control connection is reconnected successfully in first time.

Syntax

```
OnConnectionRecovery( LONG nID, BSTR szMediaSourceName )
```

Parameters

Name	Type	Description
nID	**LONG**	Return camera ID
szMediaSourceName	**BSTR**	Return media source name

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnConnectionRecovery()" LANGUAGE="Jscript">
    Alert( "Connection Recovery" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnConnectionRecovery()
        MsgBox "Connection Recovery"
    End Function
</SCRIPT>
```

See Also

OnDisconnect()

Description

Event generated when the control disconnect .

Syntax

```
OnDisconnect( LONG m_nID )
```

Parameters

Name	*Type*	*Description*
m_nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT=”OnDisconnect()” LANGUAGE=”Jscript”>
    Alert( “Disconnect !!” );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnDisconnect()
        MsgBox “Disconnect”
    End Function
</SCRIPT>
```

See Also

OnMouseKeyDoubleClick()

Description

Event generated when the user double clicks on the media area in the media window.

Syntax

`OnMouseKeyDoubleClick( LONG nID, LONG nBtn, LONG nX, LONG nY )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nBtn	LONG	Determine which mouse button is clicked. 1 is mouse double click. 2 is scroll double click. 3 is right double click.
nX	LONG	X coordinate of the click window, this value will be passed to the hosting application
nY	LONG	Y coordinate of the click window, this value will be passed to the hosting application

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT FOR=Media EVENT=”OnClickInContent(x,y)” LANGUAGE=”Jscript”>
    Alert( “Click in content: X(“ + x + “) y(“ + y + “)” );
</SCRIPT>

‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnClickInContent(x,y)
        MsgBox “Click in content: X(“ + x + “) y(“ + y + “)”
    End Function
</SCRIPT>
```

See Also

`OnMouseKeyDown(), OnMouseKeyUp(), OnMouseMoveNotify()`

OnMouseKeyDown()

Description

Event generated when the mouse down on the media area in the media window.

Syntax

```
OnMouseKeyDown( LONG nID, LONG nBtn, LONG nX, LONG nY )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nBtn	LONG	Determine which mouse button is clicked.
nX	LONG	X coordinate of the click window, this value will be passed to the hosting application
nY	LONG	Y coordinate of the click window, this value will be passed to the hosting application

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnmouseKeyDowm(ID, BTN, x,y)" LANGUAGE="Jscript">
    Alert( "Click in content: ID(" + ID + ")BTN(" + BTN + ")X(" + x + ") y(" +
y + ")" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_ OnmouseKeyDowm (ID, BTN, x,y)
        MsgBox "Click in content: ID(" + ID + ")BTN(" + BTN + ")X(" + x + ")
y(" + y + ")"
    End Function
</SCRIPT>
```

See Also

OnMouseKeyDoubleClick(), OnMouseKeyUp(), OnMouseMoveNotify()

OnMouseKeyUp()

Description

Event generated when the mouse up on the media area in the media window.

Syntax

`OnMouseKeyUp( LONG nID, LONG nBtn, LONG nX, LONG nY )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return Camera ID
nBtn	LONG	Determine which mouse button is clicked.
nX	LONG	X coordinate of the click window, this value will be passed to the hosting application
nY	LONG	Y coordinate of the click window, this value will be passed to the hosting application

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT FOR=Media EVENT=”OnmouseKeyUp(ID, BTN, x, y)” LANGUAGE=”Jscript”>
    Alert( “Click in content:ID(” + ID + “)BTN(” + BTN + “)X(“ + x + “) y(“ +
y + “)” );
</SCRIPT>
 ‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_ OnmouseKeyUp (ID, BTN, x,y)
        MsgBox “Click in content: ID(“ + ID + “)BTN(” + BTN + “)X(“ + x + “)
y(“ + y + “)”
    End Function
</SCRIPT>
```

See Also

`OnMouseKeyDoubleClick()`, `OnMouseKeyDown()`, `OnMouseMoveNotify()`

OnMouseMoveNotify()

Description

Event generated when the mouse left button down and move on the media area in the media window.

Syntax

`OnMouseMoveNotify( LONG nID, LONG nX, LONG nY )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nX	LONG	X coordinate of the click window, this value will be passed to the hosting application
nY	LONG	Y coordinate of the click window, this value will be passed to the hosting application

Returns

NONE

Remarks

To use this event, call objStreaming.EnableMoudeMoveEvent(1) first.

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT=”OnMouseMoveNotify(id, x,y)” LANGUAGE=”Jscript”>
    Alert( “Click in content: ID(“ + id + ”)X(“ + x + “) y(“ + y + “)” );
</SCRIPT>
```

See Also

OnMouseKeyDoubleClick(), OnMouseKeyDown(), OnMouseKeyUp(),

EnableMoudeMoveEvent()

OnNetworkLoss()

Description

Event generated when the connection break.

Syntax

```
OnNetworkLoss( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnNetworkLoss(nID)" LANGUAGE="Jscript">
    Alert( nID + " network loss" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnNetworkLoss (nID)
        MsgBox nID + " network loss"
    End Function
</SCRIPT>
```

See Also

OnPlay()

Description

Event generated when the control start preview.

Syntax

`OnPlay( LONG nID )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnConnect()" LANGUAGE="Jscript">
    Alert( "Connect Success" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnConnect()
        MsgBox "Connect Success"
    End Function
</SCRIPT>
```

See Also

Play()

OnPlaybackComplete()

Description

Event generated when the control playback file complete.

Syntax

```
OnPlaybackComplete( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnPlaybackComplete()" LANGUAGE="Jscript">
    Alert( "PlaybackComplete" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnPlaybackComplete()
        MsgBox "PlaybackComplete"
    End Function
</SCRIPT>
```

See Also

Play()

OnSaveImage()

Description

Event generated when save an image.

Syntax

```
OnSaveImage( LONG nID, LONG Successful, BSTR FileName )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
Successful	LONG	Return if save image successful
FileName	BSTR	Return image file name

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnSaveImage()" LANGUAGE="Jscript">
    Alert( "Save image!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_ OnSaveImage ()
        MsgBox "Save image!!"
    End Function
</SCRIPT>
```

See Also

OnStartRecord()

Description

Event generated when recording starts.

Syntax

```
OnStartRecord( LONG nID, LONG nSuccessful, BSTR szFileName )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nSuccessful	LONG	Return if record successful
szFileName	BSTR	Return record file name

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnStartRecord()" LANGUAGE="Jscript">
    Alert( "Start to record!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_ OnStartRecord()
        MsgBox "Start to record!!"
    End Function
</SCRIPT>
```

See Also

OnStopRecord()

OnStop()

Description

Event generated when the control stop preview.

Syntax

```
OnStop( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnConnect()" LANGUAGE="Jscript">
    Alert( "Connect Success" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnConnect()
        MsgBox "Connect Success"
    End Function
</SCRIPT>
```

See Also

OnStopRecord()

Description

Event generated when recording stops.

Syntax

`OnStopRecord( LONG nID, BSTR szFileName )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
szFileName	BSTR	Return record file name

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnStopRecord()" LANGUAGE="Jscript">
    Alert( "Record Stop!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnStopRecord()
        MsgBox "Record Stop!!"
    End Function
</SCRIPT>
```

See Also

`StartRecord(), StopRecord()`

OnTimeCode()

Description

Event generated when after start media stream and time code updated.

Syntax

`OnTimeCode( LONG nID, BSTR szTimeCode )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
szTimeCode	BSTR	Return camera time

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT=" OnTimeCode(nID, szTimeCode)" LANGUAGE="Jscript">
    Alert( "Record Stop!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_ OnTimeCode ()
        MsgBox "Record Stop!!"
    End Function
</SCRIPT>
```

See Also

StartRecord(), StopRecord()

OnVideoLoss()

Description

Event generated when video loss occurs .

Syntax

```
OnVideLoss( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnVideoLoss()" LANGUAGE="Jscript">
    Alert( "Video Loss!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnVideoLoss))
        MsgBox "Video Loss!!"
    End Function
</SCRIPT>
```

See Also

`OnVideoRecovery()`

OnVideoRecovery()

Description

Event generated when video recovery occurs after video loss.

Syntax

```
OnVideRecovery( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnVideoRecovery()" LANGUAGE="Jscript">
    Alert( "Video Recovery!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnVideoRecovery))
        MsgBox "Video Recovery!!"
    End Function
</SCRIPT>
```

See Also

`OnVideoLoss()`

OnKeyDown()

Description

Event generated when the keyboard down.

Syntax

```
OnKeyDown( LONG nID, LONG key )
```

Parameters

Name	***Type***	***Description***
nID	LONG	Return camera ID
key	LONG	Determine which key is press down.

Returns

NONE

Remarks

NONE

OnKeyUp()

Description

Event generated when the keyboard up.

Syntax

```
OnKeyUp( LONG nID, LONG key )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
key	LONG	Determine which key is press up.

Returns

NONE

Remarks

NONE

Motion Detection Events

Name	*Description*
OnMDEventStart()	Event triggered when a motion detection event occurs
OnMDEventEnd()	Event triggered when a motion detection event End
OnMDSetting()	Event generated when motion detection setting is enabled
OnMDSettingChange()	Event generated when motion detection setting be changed

OnMDEventStart()

Description

Event generated when motion detection event occurs.

Syntax

`OnMDEventStart( LONG nID, LONG nMD )`

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nMD	LONG	The sum of regional motion detection numbers triggered.MD1 = 1, MD2 = 2, MD3 = 4. If MD1 and MD3 triggered, the number is 5.

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnMDEventStart(SHORT nMD)" LANGUAGE="JScript">
    Alert( "MD Event: Range " + nMD + " detected !" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnMDEventStart(SHORT nMD)
        MsgBox "MD Event: Range " + nMD + " detected !!"
    End Function
</SCRIPT>
```

See Also

EnableMotionDetection(), OnMDEventEnd(), MotionDetectionAlertDuration,

OnMDEventEnd()

Description

Event generated when motion detection event end.

Syntax

`OnMDEventEnd( LONG *nID* )`

Parameters

Name	*Type*	*Description*
nID	`LONG`	Return camera ID

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnMDEventEnd()" LANGUAGE="JScript">
    Alert( "MD Event End !" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnMDEventEnd
        MsgBox "MD Event End !!"
    End Function
</SCRIPT>
```

See Also

`EnableMotionDetection(), OnMDEventStart(), MotionDetectionAlertDuration,`

OnMDSetting()

Description

Event generated when motion detection setting is enabled.

Syntax

```
OnMDSetting ( LONG nID,
              LONG nIndex,
              LONG nX,
              LONG nY,
              LONG nW,
              LONG nH,
              LONG nSens,
              LONG enable )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nIndex	LONG	Return MD index
nX	LONG	Return MD start position in x-axis
nY	LONG	Return MD start position in y-axis
nW	LONG	Return MD width
nH	LONG	Return MD height
nSens	LONG	Return MD sensitive
enable	LONG	Return MD enable status

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT FOR=Media EVENT=” OnMDSetting()” LANGUAGE=”Jscript”>
    Alert( “Setting the motion!!” );
</SCRIPT>

‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_ OnMDSetting()
        MsgBox “Setting the motion!!”
    End Function
</SCRIPT>
```

See Also

OnMDSettingChange()

Description

Event generated when motion detection setting be changed.

Syntax

```
OnMDSettingChange( LONG nID, LONG nIndex )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nIndex	LONG	Return MD index

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT=" OnMDSettingChange()" LANGUAGE="Jscript">
    Alert( "Changing the motion setting!!" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_ OnMDSettingChange()
        MsgBox "Changing the motion setting!!"
    End Function
</SCRIPT>
```

See Also

DIO Events

Name	*Description*
OnDIEvent()	Event generated when the digital in port is triggered.
OnDIEvent()	Event generated when the digital in port is ended.
OnRS232Event()	Event generated when data comes from RS-232/RS-422/RS-485 port

OnDIEvent()

Description

Event generated when the digital in port is triggered.

Syntax

```
OnDIEvent( LONG nID, LONG nDI )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
nDI	LONG	Indicates DI port number that is triggered

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnDIEvent(SHORT nDI)" LANGUAGE="JScript">
    Alert( "DI Event: " + nDI + " fired !" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnDIEvent(SHORT nDI)
        MsgBox "DI Event: " + nDI + " fired !!"
    End Function
</SCRIPT>
```

See Also

OnDIEventEnd()

Description

Event generated when the digital in port is ended.

Syntax

```
OnDIEvent( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnDIEvent(SHORT nDI)" LANGUAGE="JScript">
    Alert( "DI Event: " + nDI + " fired !" );
</SCRIPT>

'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnDIEvent(SHORT nDI)
        MsgBox "DI Event: " + nDI + " fired !!"
    End Function
</SCRIPT>
```

See Also

OnRS232Event()

Description

Event generated when data comes from RS-232/RS-422/RS-485 port.

Syntax

```
OnRS232Event( LONG nID, BSTR szData, LONG nDataLen )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera ID
szData	BSTR	The data received from the RS-232/RS-422/RS-485 port
nDataLen	LONG	The length of the data received

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT FOR=Media EVENT=”OnRS232Event(RS232_Data, iLength)” LANGUAGE=”JScript”>
    Alert( “RS232 Data: “ + RS232_Data + “ , len = ” + iLength + ”!” );
</SCRIPT>

‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnRS232Event(nID, RS232_Data, iLength)
        MsgBox “RS232 Data: “ + RS232_Data + “ , len = ” + iLength + ”!”
    End Function
</SCRIPT>
```

See Also

5 New Method

This section lists new implement method.

New Method

Name	*Description*
SetTextOut()	Set text show on control display screen
MirrorImage()	Reverse image from left to right
FlipImage()	Reverse image for top to bottom
EnablePrivacyMask()	Enable privacy mask function and set the mask color
SetPrivacyMask()	Set privacy mask region
EnableAbsPosition()	Enable absolute PTZ position function
DisableAbsPosition()	Disable absolute PTZ position function
AddMultiplePlaybackFile()	Add file to multiple playback file list.
RemoveAllMultiplePlaybackFile()	Remove all file in the multiple playback file list.
PlayAudioFileToDevice()	Play audio file to device.
GetNumberOfMonitor()	Get number of monitors in os
ShowRealFPS()	Show real fps number on screen
SetRealFpsFontColor()	Set font and color for show real fps
SetTextOutEx()	Set text show on control display screen
ExecuteURLCommand()	Execute URL command
SendURLCmd()	Send URL command
SendPTZCommand()	Send PTZ protocol command to video server or speed dome
RequestsToCheckDeviceChipType()	Request to check device's chip type
EnableFullScreenEx()	Enable the Fullscreen
TxRS232Data()	Send RS232 command
AsyncDisconnect()	Disconnect the media source asynchronously
AsyncSendURLCmd()	Send URL command asynchronously
EnableOnNewImageEvent()	Enable OnMyNewImage() event
SetGlobalMotionEnable()	Set global motion parameter enable or disable
GetGlobalMotionEnable()	Get global motion parameter is enable or disable
SnapshotWithTitleBar()	SnapShot with title bar
GetFullScreenStatus()	Get fullscreen status

CloseFullScreenWindow()	Close the fullscreen mode
GetMouseDigitalPTZStatus()	Get mouse digital PTZ status
GetPlayRate()	Get playback speed
GetPlayDirection()	Get the direction of playback forward or backward
SetUtcCurrentTime()	Sets current time in playback mode use UTC time
SetTextOutToRegion()	Set text show on control display screen
StartRecordEx()	Start play current media stream or media file
StopRecordEx()	Stops recording on the current recording operation
SnapShotEx()	Saves current SnapShot image of current video frame by identity filename
GetCurrentTimeByInteger()	Gets current time in playback mode
SetCurrentTimeByInteger()	Sets current time in playback mode
ConvertUTCtoString()	Convert UTC time to string
EnablePIRMotionDetection()	Enable PIR’s motion detection
DisablePIRMotionDetection()	Disable PIR’s motion detection
GetPIRConfig()	Get PIR settings
SetPIRConfig()	Set PIR settings
SetMotionTime()	Set the motion timer to the specific motion region
EnableMotionSetting()	Enable to setting motion detection settings
SetMotionSettingEx()	Set motion detection settings
GetMotionInfoFromDevice()	Get motion detection information from device
StartStreamEx()	Starts Decoding mpeg4 stream but don’t show in the window
SetDigitalOutput()	Set digital out event enable
DevicePrivacyMask()	Enable/Disable device privacy mask
SnapShotEx4CMS()	Saves current SnapShot image of current video frame by identity filename
GetCurrentImage()	Get current image
GetCurrentImageEx()	Get current image
EnableDecoder()	Set decoder enable or disable
EnableMouseMoveEvent()	Set mouse move event enable or disable
SendRS232Command()	Send RS232 command
RefreshControl()	Refresh control
EnableTitleEvent()	Set title event enable
DisableTitleEvent()	Set title event disable

SetTextOut()

Description

Set text show on control display screen

Syntax

```
LONG SetTextOut( LONG nX,
                  LONG nY,
                  BSTR szText,
                  BSTR szFontName,
                  SHORT nBold,
                  LONG nFontWidth,
                  LONG nnFontHeight,
                  LONG nFontR,
                  LONG nFontG,
                  LONG nFontB )
```

Parameters

Name	*Type*	*Description*
nX	Long	X coordinate of text
nY	Long	Y coordinate of text
szText	BSTR	String that want to show on screen
szFontName	BSTR	Font name of showed text
nBold	SHORT	Font weight setting.
nFontWidth	LONG	Width of text
nFontHeight	LONG	Height of text
nFontR	LONG	Red in text color setting
nFontG	LONG	Green in text color setting
nFontB	LONG	Blue in text color setting

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result. Value / Description 0 Fail 1 Success

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......
objStreaming.Play()
......
objStreaming.SetTextOut(5,5,”Text out String”,”Arial”,0,10,20,255,255,255) ;
......
```

See Also

NONE

MirrorImage()

Description

Reverse image from left to right.

Syntax

```
MirrorImage( LONG nEnable )
```

Parameters

Name	*Type*	*Description*
nEnable	LONG	Set 0 for disable 1 for enable function

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.Play()
......
objStreaming.MirrorImage(0) ;
......
```

See Also

NONE

FlipImage()

Description

Reverse image from up to down.

Syntax

```
FlipImage( LONG nEnable )
```

Parameters

Name	*Type*	*Description*
nEnable	LONG	Set 0 for disable, 1 for enable function

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.Play()
......
objStreaming.FlipImage(1) ;
......
```

See Also

NONE

EnablePrivacyMask()

Description

Enable privacy mask function and set mask color

Syntax

```
EnablePrivacyMask( LONG nEnable, LONG R, LONG G, LONG B )
```

Parameters

Name	*Type*	*Description*
nEnable	Long	Set 0 for disable, 1 for enable function
R	Long	Red in RGB color
G	Long	Green in RGB color
B	Long	Blue in RGB color

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.Play()
......
objStreaming.EnablePrivacyMask(1,255,255,255) ;
......
```

See Also

NONE

SetPrivacyMask()

Description

Set privacy mask region

Syntax

```
SetPrivacyMask( LONG nIndex, LONG nXStart, LONG nYStart, LONG nXEnd, LONG nYEnd)
```

Parameters

Name	*Type*	*Description*
nIndex	LONG	Privacy mask index(1~3)
nXStart	LONG	X coordinate of left-top corner
nYStart	LONG	Y coordinate of left-top corner
nXEnd	LONG	X coordinate of right-bottom corner
nYEnd	LONG	Y coordinate of right-bottom corner

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.Play()
......
objStreaming.SetPrivacyMask(3,50,50,150,150) ;
......
```

See Also

NONE

EnableAbsPosition()

Description

Enable PTZ absolute position function

Syntax

```
LONG EnableAbsPosition()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result. **Value** / **Description** **0** / Fail **1** / Success

Remarks

NONE

Example

```
......
objStreaming.EnablePTZ() ;
......
objStreaming.EnableAbsPosition() ;
......
```

See Also

NONE

DisableAbsPosition()

Description

Disable PTZ absolute position function

Syntax

```
DisableAbsPosition()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.DisableAbsPosition() ;
......
```

See Also

NONE

AddMultiplePlaybackFile()

Description

Add file to multiple playback file list.

Syntax

```
LONG AddMultiplePlaybackFile( BSTR szFileName )
```

Parameters

Name	*Type*	*Description*
szFileName	BSTR	Full path of playback file.(It must has idx file in the same folder)

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Fail</td></tr>
<tr><td><b>1</b></td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......
objStreaming.MediaType = 6
......
objStreaming.AddMultiplePlaybackFile(“C:\RecTest1.RAW”)
objStreaming.AddMultiplePlaybackFile(“C:\RecTest2.RAW”)
......
```

See Also

NONE

RemoveAllMultiplePlaybackFile()

Description

Remove all file in the multiple playback file list.

Syntax

```
RemoveAllMultiplePlaybackFile()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
objStreaming.MediaType = 6
......
objStreaming.RemoveAllMultiplePlaybackFile()
......
```

See Also

NONE

PlayAudioFileToDevice()

Description

Play the audio file to device.

Syntax

```
LONG PlayAudioFileToDevice( BSTR szFileName )
```

Parameters

Name	*Type*	*Description*
szFileName	BSTR	Full path of audio file.(It must be 8k 16bit PCM format)

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......
objStreaming.MediaType = 6
......
objStreaming.PlayAudioFileToDevice(“C:\Audio.wav”)
......
```

See Also

StopPlayingAudioFile()

GetNumberOfMonitors()

Description

Get number of monitors in os.

Syntax

```
LONG GetNumberOfMonitors()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Return number of monitors.

Remarks

NONE

ShowRealFPS()

Description

Show real fps number on screen.

Syntax

```
ShowRealFPS( SHORT bShow )
```

Parameters

Name	*Type*	*Description*
bShow	`SHORT`	1: visible, 2: hiden.

Returns

NONE

Remarks

NONE

SetRealFpsFontColor()

Description

Set font and color for show real fps.

Syntax

```
SetRealFpsFontColor( LONG bBold,
                     LONG bItalic,
                     LONG bUnderLine,
                     BSTR pFontName,
                     LONG nFontSize,
                     LONG nTextColorR,
                     LONG nTextColorG,
                     LONG nTextColorB,
                     LONG nBKMode,
                     LONG nBkR,
                     LONG nBkG,
                     LONG nBkB )
```

Parameters

Name	*Type*	*Description*
bBold	LONG	Is bold.
bItalic	LONG	Is italic.
bUnderLine	LONG	Is underline.
pFontName	BSTR	Font typeface name.
nFontSize	LONG	Font size.
nTextColorR	LONG	Text color R.
nTextColorG	LONG	Text color G.
nTextColorB	LONG	Text color B.
nBKMode	LONG	Specifies the background mode.
nBkR	LONG	Background color R.
nBkG	LONG	Background color G.
nBkB	LONG	Background color B.

Returns

NONE

Remarks

NONE

SetTextOutEx()

Description

Set font and color for show real fps.

Syntax

```
SetTextOutEx( LONG index,
              LONG nX,
              LONG nY,
              BSTR Text,
              LONG bBold,
              LONG bItalic,
              LONG bUnderLine,
              BSTR pFontName,
              LONG nFontSize,
              LONG nTextColorR,
              LONG nTextColorG,
              LONG nTextColorB,
              LONG nBKMode,
              LONG nBkR,
              LONG nBkG,
              LONG nBkB )
```

Parameters

Name	***Type***	***Description***
index	LONG	Specify a index to store data, maximum is 10.
nX	LONG	x-coordinate of pixel.
nY	LONG	y-coordinate of pixel.
Text	BSTR	String, maximum is 1024
bBold	LONG	Is bold.
bItalic	LONG	Is italic.
bUnderLine	LONG	Is underline.
pFontName	BSTR	Font typeface name.
nFontSize	LONG	Font size.
nTextColorR	LONG	Text color R.
nTextColorG	LONG	Text color G.
nTextColorB	LONG	Text color B.
nBKMode	LONG	Specifies the background mode.
nBkR	LONG	Background color R.

nBkG	**LONG**	Background color G.
nBkB	**LONG**	Background color B.

Returns

NONE

Remarks

NONE

ExecuteURLCommand ()

Description

Execute URL command.

Syntax

```
LONG ExecuteURLCommand( BSTR szCommand )
```

Parameters

Name	*Type*	*Description*
szCommand	BSTR	URL command.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Return zero.

Remarks

NONE

Example

```
......
......
objStreaming.ExecuteURLCommand("http://192.168.1.2");
......
```

See Also

SendURLCmd ()

Description

Send URL command.

Syntax

```
BSTR SendURLCmd( BSTR szCommand )
```

Parameters

Name	*Type*	*Description*
szcommand	BSTR	URL command.

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Return setting result.

Remarks

NONE

Example

```
......
......
objStreaming. SendURLCmd (“http://192.168.1.2”);
......
```

See Also

SendPTZCommand ()

Description

Send PTZ protocol command to video server or speed dome.

Syntax

`SendPTZCommand( BSTR szHEXStringCommand )`

Parameters

Name	*Type*	*Description*
szHEXStringCommand	BSTR	The PTZ command that is defined by protocol file.

Returns

NONE

Remarks

NONE

RequestsToCheckDeviceChipType()

Description

Request to check device’s chip type.

Syntax

```
RequestsToCheckDeviceChipType( LONG nCheckIt )
```

Parameters

Name	*Type*	*Description*
nCheckIt	LONG	0: no check 1: check

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. RequestsToCheckDeviceChipType (1);
......
```

See Also

EnableFullScreenEx()

Description

Enable the Fullscreen and set to second monitor or not in multi-monitor mode.

Syntax

```
EnableFullScreenEx( LONG nUseSecondMonitor )
```

Parameters

Name	*Type*	*Description*
nUseSecondMonitor	LONG	0: Disable 1: Enable.

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. EnableFullScreenEx (1);
......
```

See Also

TxRS232Data()

Description

Send RS232 command.

Syntax

```
LONG TxRS232Data( OLE_HANDLE pData, LONG numBytes )
```

Parameters

Name	*Type*	*Description*
pData	OLE_HANDLE	The handle of video.
numBytes	LONG	Command length.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns setting result.

Value	Description
0	Success
-1	Media type must be : PREVIEW, PREVIEW_RTSP, PREVIEW_RTSP_WOC, MULTICAST, PREVIEW_AHTTP, PREVIEW_AHTTP_PREVIEW, PREVIEW_AHTTP_WITHOUT_CONTROL, PREVIEW_AHTTP_CONTROL_ONLY.
-2	Data Len size <=0 or > 4096

Remarks

NONE

Example

```
......
......
objStreaming. TxRS232Data (handle,2048);
......
```

See Also

AsyncDisconnect()

Description

Disconnect the media source asynchronously.

Syntax

```
LONG AsyncDisconnect()
```

Parameters

NONE

Returns

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>Return_Value</em></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><strong>0</strong></td><td>Fail</td></tr>
<tr><td><strong>1</strong></td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......
......
objStreaming. AsyncDisconnect ();
......
```

See Also

AsyncSendURLCmd()

Description

Send URL command asynchronously.

Syntax

```
LONG AsyncSendURLCmd( LONG nIndex, BSTR szCommand )
```

Parameters

Name	*Type*	*Description*
nIndex	LONG	0: hiden 1: visible.
szCommand	BSTR	URL Command.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......
......
objStreaming. AsyncSendURLCmd (1);
......
```

See Also

EnableOnNewImageEvent()

Description

Enable OnMyNewImage() event.

Syntax

```
LONG EnableOnNewImageEvent( LONG nEnable )
```

Parameters

Name	*Type*	*Description*
nEnable	LONG	0: disable 1: enable.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns zero.

Remarks

NONE

Example

```
......
......
objStreaming. EnableOnNewImageEvent (1);
......
```

See Also

SetGlobalMotionEnable()

Description

Set global motion parameter enable or disable.

Syntax

```
SetGlobalMotionEnable( ULONG bEnable )
```

Parameters

Name	*Type*	*Description*
bEnable	ULONG	0: Disable 1: Enable.

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SetGlobalMotionEnable (1);
......
```

See Also

GetGlobalMotionEnable()

Description

Get global motion parameter is enable or disable.

Syntax

```
ULONG GetGlobalMotionEnable()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Return setting value.

Remarks

NONE

Example

```
......
......
int return_val = objStreaming. GetGlobalMotionEnable();
......
```

See Also

SnapshotWithTitleBar ()

Description

Snapshot with title bar.

Syntax

```
SnapshotWithTitleBar( ULONG bEnable )
```

Parameters

Name	*Type*	*Description*
bEnable	ULONG	0: Disable 1: Enable.

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SnapshotWithTitleBar(1);
......
```

See Also

GetFullScreenStatus()

Description

Get fullscreen status.

Syntax

```
LONG GetFullScreenStatus()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	0: Normal mode 1: Fullscreen mode.

Remarks

NONE

Example

```
......
......
int val = objStreaming. GetFullScreenStatus ();
......
```

See Also

CloseFullScreenWindow()

Description

Close the fullscreen mode.

Syntax

```
CloseFullScreenWindow()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. CloseFullScreenWindow ();
......
```

See Also

GetMouseDigitalPTZStatus()

Description

Get mouse digital PTZ status.

Syntax

```
LONG GetMouseDigitalPTZStatus()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	0: Disable 1: Enable.

Remarks

NONE

Example

```
......
......
long val = objStreaming. GetMouseDigitalPTZStatus ();
......
```

See Also

GetPlayRate()

Description

Get playback speed.

Syntax

```
LONG GetPlayRate()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Playback speed.

Remarks

NONE

Example

```
......
......
long val = objStreaming. GetPlayRate ();
......
```

See Also

GetPlayDirection()

Description

Get the direction of playback forward or backward.

Syntax

```
LONG GetPlayDirection()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	0: backward 1: forward.

Remarks

NONE

Example

```
......
......
long val = objStreaming. GetPlayDirection ();
......
```

See Also

SetUtcCurrentTime ()

Description

Sets current time in playback mode use UTC time.

Syntax

```
SetUtcCurrentTime( LONG val )
```

Parameters

Name	*Type*	*Description*
val	`LONG`	Current UTC time.

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SetUtcCurrentTime (123456);
......
```

See Also

SetTextOutToRegion()

Description

Set text show on control display screen.

Syntax

```
SetTextOutToRegion( LONG index,
                    LONG nRegion,
                    BSTR Text,
                    LONG bBold,
                    LONG bItalic,
                    LONG bUnderLine,
                    BSTR pFontName,
                    LONG nFontSize,
                    LONG nTextColorR,
                    LONG nTextColorG,
                    LONG nTextColorB,
                    LONG nBKMode,
                    LONG nBkR,
                    LONG nBkG,
                    LONG nBkB  )
```

Parameters

Name	*Type*	*Description*
index	LONG	Line number is range from 0~15.
nRegion	LONG	Line number is range from 0~15.
Text	BSTR	Title text
bBold	LONG	0:Non Bold 1:Bold.
bItalic	LONG	0:Non Italic 1:Italic.
bUnderLine	LONG	0:Non UnderLine 1: UnderLine.
pFontName	BSTR	Set Font style.
nFontSize	LONG	Set Font size.
nTextColorR	LONG	Text color(0~255).
nTextColorG	LONG	Text color(0~255).
nTextColorB	LONG	Text color(0~255).
nBKMode	LONG	Background mode.
nBkR	LONG	Background color(0~255).
nBkG	LONG	Background color(0~255).
nBkB	LONG	Background color(0~255).

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SetTextOutToRegion (1, 1, "Test", 0, 0,0, "Arial", 20, 255, 255,
255, 3, 0, 0, 0);
......
```

See Also

StartRecordEx()

Description

Start recording current media stream or media file.

Syntax

```
LONG StartRecordEx( BSTR szFileFolder )
```

Parameters

Name	*Type*	*Description*
szFileFolder	BSTR	Set recordind file saving folder.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Fail</td></tr>
<tr><td>1</td><td>Success</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

```
......
......
objStreaming. StartRecordEx (“C://”);
......
```

See Also

StopRecordEx()

Description

Stops recording on the current recording operation.

Syntax

```
StopRecordEx()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. StopRecordEx ();
......
```

See Also

SnapShotEx()

Description

Saves current SnapShot image of current video frame by identity filename.

Syntax

```
BSTR SnapShotEx( LONG IFormat,
                 BSTR FileFolder,
                 LONG ReduplicateTitleBarText,
                 LONG R_TextColor,
                 LONG G_TextColor,
                 LONG B_TextColor )
```

Parameters

Name	*Type*	*Description*
IFormat	LONG	0:BMP 1:JPG 2:GIF 3:TIFF 4:PNG.
FileFolder	BSTR	Set SnapShot file saving folder.
ReduplicateTitleBarText	LONG	Specifies the word to the picture reduplicate.
R_TextColor	LONG	Specifies the red color to the picture adjust.
G_TextColor	LONG	Specifies the green color to the picture adjust.
B_TextColor	LONG	Specifies the blue color to the picture adjust.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns null.

Remarks

NONE

Example

```
......
......
objStreaming. SnapShotEx (1, "c:\RecTest\SnapShot" & nFile & ".jpg", 1, r_Color,
g_color, b_color);
......
```

See Also

GetCurrentTimeByInteger()

Description

Gets current time in playback mode.

Syntax

```
LONG GetCurrentTimeByInteger()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Current time.

Remarks

NONE

Example

```
......
......
int val = objStreaming. GetCurrentTimeByInteger ();
......
```

See Also

SetCurrentTimeByInteger()

Description

Sets current time in playback mode.

Syntax

```
SetCurrentTimeByInteger( LONG nCurrentTime )
```

Parameters

Name	*Type*	*Description*
nCurrentTime	LONG	Set currect time by integer.

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SetCurrentTimeByInteger (123456);
......
```

See Also

ConvertUTCtoString()

Description

Convert UTC time to string.

Syntax

```
BSTR ConvertUTCtoString( LONG nTimeStamp, BSTR szFormat )
```

Parameters

Name	*Type*	*Description*
nTimeStamp	LONG	Time stamp.
szFormat	BSTR	Time Format.

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Time string

Remarks

NONE

Example

```
......
......
objStreaming. ConvertUTCtoString (123456,”UTC”);
......
```

See Also

EnablePIRMotionDetection()

Description

Enable PIR’s motion detection.

Syntax

```
EnablePIRMotionDetection()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. EnablePIRMotionDetection ();
......
```

See Also

DisablePIRMotionDetection()

Description

Disable PIR’s motion detection.

Syntax

```
DisablePIRMotionDetection()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. DisablePIRMotionDetection ();
......
```

See Also

GetPIRConfig()

Description

Get PIR settings.

Syntax

```
BSTR GetPIRConfig()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Success: "bEnable, dwSensitive, dwTime". Fail: "0,0,0"

Remarks

NONE

Example

```
......
......
BSTR str = objStreaming. GetPIRConfig ();
......
```

See Also

SetPIRConfig()

Description

Set PIR settings.

Syntax

```
ULONG SetPIRConfig( ULONG bEnable, ULONG nSensitive, ULONG nTime )
```

Parameters

Name	*Type*	*Description*
bEnable	ULONG	0: Disable 1: Enable.
nSensitive	ULONG	current motion sensitive value.
nTime	ULONG	current motion timer value.

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Returns setting result.

Value	Description
0	Fail
1	Success

Remarks

NONE

Example

```
......
......
objStreaming. SetPIRConfig (1,80,10);
......
```

See Also

SetMotionTime()

Description

Set the motion timer to the specific motion region.

Syntax

```
SetMotionTime( LONG nMotionRegionNumber, LONG nTime )
```

Parameters

Name	*Type*	*Description*
nMotionRegionNumber	LONG	Set the special the motion region number.
nTime	LONG	current motion timer value.

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SetMotionTime (1,10);
......
```

See Also

EnableMotionSetting()

Description

Enable to setting motion detection settings.

Syntax

```
ULONG EnableMotionSetting( ULONG bEnable )
```

Parameters

Name	*Type*	*Description*
bEnable	ULONG	0: Disable 1: Enable.

Returns

Name	*Type*	*Description*
Return_Value	ULONG	Return one.

Remarks

NONE

Example

```
......
......
objStreaming. EnableMotionSetting (1);
......
```

See Also

SetMotionSettingEx()

Description

Set motion detection settings.

Syntax

```
SetMotionSettingEx( LONG nMotionRegionNumber,
                    LONG nStartX,
                    LONG nStartY,
                    LONG nEndX,
                    LONG nEndY,
                    LONG nSensitive,
                    LONG nTime,
                    LONG bEnable )
```

Parameters

Name	*Type*	*Description*
nMotionRegionNumber	LONG	Motion detection window number. Valid values are: 1, 2, 3
nStartX	LONG	Motion detection window region starting X value
nStartY	LONG	Motion detection window region starting Y value
nEndX	LONG	Motion detection window region ending X value
nEndY	LONG	Motion detection window region ending Y value
nSensitive	LONG	Motion detection window region sensitivity
nTime	LONG	Motion detection window region timer
bEnable	LONG	Motion detection window region enable

Returns

NONE

Remarks

NONE

Example

```
objStreaming. SetMotionSettingEx (1,0,0,100,100,80,10,1);
```

See Also

GetMotionInfoFromDevice()

Description

Get motion detection information from device.

Syntax

```
BSTR GetMotionInfoFromDevice()
```

Parameters

NONE

Returns

Name	*Type*	*Description*

Return_Value	BSTR	MOTION_ENABLED=”0x01”
		MOTION_CONFIG=”Region ,number,bEnable, StartX,StartY,EndX,EndY, Sensitive,Timer, Threshold”

Remarks

NONE

Example

```
......
......
BSTR str = objStreaming. GetMotionInfoFromDevice ();
......
```

See Also

StartStreamEx()

Description

Starts Decoding mpeg4 stream but don’t show in the window.

Syntax

```
LONG StartStreamEx()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	0: fail 1: success.

Remarks

NONE

Example

```
......
......
long val = objStreaming. StartStreamEx ();
......
```

See Also

SetDigitalOutput()

Description

Set digital out event enable.

Syntax

```
SetDigitalOutput( LONG nDOIndex, LONG enable )
```

Parameters

Name	*Type*	*Description*
nDOIndex	LONG	DI index.
enable	LONG	0:disable 1:enable

Returns

NONE

Remarks

NONE

Example

```
......
......
objStreaming. SetDigitalOutput (1,1);
......
```

See Also

DevicePrivacyMask()

Description

Enable/Disable device privacy mask.

Syntax

```
BSTR DevicePrivacyMask( LONG nEnable )
```

Parameters

Name	*Type*	*Description*
nEnable	LONG	0: disable 1: enable.

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Return null.

Remarks

NONE

Example

```
......
......
objStreaming. DevicePrivacyMask (1);
......
```

See Also

SnapShotEx4CMS()

Description

Saves current SnapShot image of current video frame by identity filename.

Syntax

```
BSTR SnapShotEx4CMS( LONG IFormat,
                     LPCTSTR FileFolder,
                     LONG ReduplicateTitleBarText,
                     LONG R_TextColor,
                     LONG G_TextColor,
                     LONG B_TextColor,
                     LPCTSTR NVRName,
                     LPCTSTR CamName )
```

Parameters

Name	*Type*	*Description*
IFormat	LONG	0:BMP 1:JPG 2:GIF 3:TIFF 4:PNG.
FileFolder	LPCTSTR	Set SnapShot file saving folder.
ReduplicateTitleBarText	LONG	Specifies the word to the picture reduplicate.
R_TextColor	LONG	Specifies the red color to the picture adjust.
G_TextColor	LONG	Specifies the green color to the picture adjust.
B_TextColor	LONG	Specifies the blue color to the picture adjust.
NVRName	LPCTSTR	NVR name
CamName	LPCTSTR	Cam name

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Return null.

Remarks

This function is for CMS’s SnapShot.

Example

```
......
......
objStreaming. SnapShotEx4CMS (1, "c:\RecTest\SnapShot" & nFile & ".jpg", 1,
r_Color, g_color, b_color,”NVR1”,”Cam1”);
......
```

See Also

GetCurrentImage()

Description

Get current image.

Syntax

```
LONG GetCurrentImage( BYTE* pBuffer, LONG nBufferLen )
```

Parameters

Name	*Type*	*Description*
pBuffer	BYTE*	Save image buffer.
nBufferLen	LONG	Image size.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Fail</td></tr>
<tr><td><b>else</b></td><td>The image size.</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

See Also

GetCurrentImageEx()

Description

Get current image.

Syntax

```
LONG GetCurrentImageEx( OLE_HANDLE Int32BufferPointer, LONG nBufferLen )
```

Parameters

Name	*Type*	*Description*
Int32BufferPointer	OLE_HANDLE	Save image buffer.
nBufferLen	LONG	Image size.

Returns

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>Return_Value</i></td><td>LONG</td><td>Returns setting result.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Fail</td></tr>
<tr><td><b>else</b></td><td>The image size.</td></tr>
</table>
</td></tr>
</table>

Remarks

NONE

Example

See Also

EnableDecoder()

Description

Set decoder enable or disable.

Syntax

```
LONG EnableDecoder( LONG nEnable )
```

Parameters

Name	*Type*	*Description*
nEnable	LONG	0: Disable, 1: Enable.

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns zero.

Remarks

NONE

Example

See Also

EnableMouseMoveEvent()

Description

Set mouse move event enable or disable.

Syntax

```
EnableMouseMoveEvent( LONG nEnable )
```

Parameters

Name	*Type*	*Description*
nEnable	`LONG`	0: Disable, 1: Enable.

Returns

NONE

Remarks

NONE

Example

See Also

SendRS232Command()

Description

Send rs232 command.

Syntax

```
SendRS232Command( LPCTSTR szCommand )
```

Parameters

Name	*Type*	*Description*
szCommand	LPCTSTR	RS232 command.

Returns

NONE

Remarks

NONE

Example

See Also

RefreshControl()

Description

Refresh control.

Syntax

```
RefreshControl()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

See Also

EnableTitleEvent()

Description

Set title event enable.

Syntax

```
LONG EnableTitleEvent()
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Return zero.

Remarks

NONE

Example

See Also

DisableTitleEvent()

Description

Set title event disable.

Syntax

```
DisableTitleEvent()
```

Parameters

NONE

Returns

NONE

Remarks

NONE

Example

See Also

6 New Property

This section lists new implement property.

New Property

Name	*Description*
MotionDetectionInterval	**[I/O]** Sets/Gets motion detection interval
RenderInterface	**[I/O]** Sets/Gets current render using interface type
PlayFileName	**[I/O]** Sets/Gets current playback file name
RealFPS	**[O]** Gets current fps value
RecordLedStatus	**[I/O]** Sets/Gets recording LED status
RecordStatus	**[O]** Gets record status
SiteID	**[I/O]** Sets/Gets site ID
PIRDetectionAlertDuration	**[I/O]** Sets/Gets PIR detection alert duration
DIDefaultValue	**[I/O]** Sets/Gets DI default value
Is4VGAMode	**[I/O]** Sets/Gets the 4VGA mode value

MotionDetectionInterval

Description

[I/O] Sets/Gets motion detection interval

Syntax

```
MotionDetectionInterval = Input_Value
Return_Value = MotionDetectionInterval [ default: 10 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets motion detection interval

Returns

Name	*Type*	*Description*
Return_Value	LONG	Gets motion detection interval

Remarks

NONE

Example

```
..
.. . .
ObjStreaming. MotionDetectionInterval = 10;
```

See Also

RenderInterface

Description

`[I/O]` Sets/Gets current render using interface type

Syntax

```
RenderInterface = Input_Value
Return_Value = RenderInterface [ default: 0 ]
```

Parameters

Name	*Type*	*Description*
`Input_Value`	`LONG`	Sets current render using interface type **Value** – **Description** **0** – DGDI **1** – DXDRAW

Returns

Name	*Type*	*Description*
`Return_Value`	`LONG`	Returns current render using interface type

Remarks

NONE

Example

See Also

PlayFileName

Description

[I/O] Sets/Gets current playback file name

Syntax

```
PlayFileName = “Input_Value”
“Return_Value” = PlayFileName
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets current playback file name

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets current playback file name

Remarks

NONE

Example

```
ObjStreaming. PlayFileName = “TEST”;
```

See Also

RealFPS

Description

[O] Gets current fps value

Syntax

```
Return_Value = RealFPS
```

Parameters

NONE

Returns

Name	Type	Description
Return_Value	ULONG	Returns current fps value

Remarks

NONE

Example

```
int val = ObjStreaming.RealFPS;
```

See Also

RecordLedStatus

Description

`[I/O]` Sets/Gets recording LED status

Syntax

```
RecordLedStatus = Input_Value
Return_Value = RecordLedStatus [ default: 0 ]
```

Parameters

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><em>Input_Value</em></td><td>LONG</td><td>Sets recording LED status
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Normal</td></tr>
<tr><td><b>1</b></td><td>RED LED</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Return_Value	LONG	Gets recording LED status

Remarks

NONE

Example

See Also

RecordStatus

Description

[o] Gets recording status

Syntax

```
Return_Value = RecordStatus
```

Parameters

NONE

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns recording status. 0: no record 1: recording

Remarks

NONE

Example

See Also

SiteID

Description

[I/O] Sets/Gets site ID

Syntax

```
SiteID = “Input_Value”
“Return_Value” = SiteID
```

Parameters

Name	*Type*	*Description*
Input_Value	BSTR	Sets site ID

Returns

Name	*Type*	*Description*
Return_Value	BSTR	Gets site ID

Remarks

NONE

Example

See Also

PIRDetectionAlertDuration

Description

`[I/O]` Sets/Gets PIR detection alert duration

Syntax

```
PIRDetectionAlertDuration = Input_Value
Return_Value = PIRDetectionAlertDuration [ default: 10 ]
```

Parameters

Name	*Type*	*Description*
Input_Value	LONG	Sets PIR detection alert duration

Returns

Name	*Type*	*Description*
Return_Value	LONG	Returns PIR detection alert duration

Remarks

NONE

Example

```
objStreaming.PIRDetectionAlertDuration = 10;
```

See Also

DIDefaultValue

Description

[I/O] Sets/Gets DI default value

Syntax

```
DIDefaultValue = Input_Value
Return_Value = DIDefaultValue [ default: 0x03 ]
```

Parameters

<table>
<tr><th><em>Name</em></th><th><em>Type</em></th><th><em>Description</em></th></tr>
<tr><td><em>Input_Value</em></td><td>BYTE</td><td>Sets the DI default value
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><strong>0x00</strong></td><td>DI_DEFAULT_IS_LOW</td></tr>
<tr><td><strong>0x03</strong></td><td>DI_DEFAULT_IS_HIGH</td></tr>
</table>
</td></tr>
</table>

Returns

Name	*Type*	*Description*
Retuen_value	BYTE	Returns the DI default value

Remarks

NONE

Example

See Also

Is4VGAMode

Description

`[I/O]` Sets/Gets the 4VGA mode value

Syntax

```
Is4VGAMode = Input_Value
Return_Value = Is4VGAMode [ default: FALSE ]
```

Parameters

Name	*Type*	*Description*
Input_Value	BOOL	Sets the 4VGA mode value

Returns

Name	*Type*	*Description*
Retuen_value	BOOL	Returns current setting value

Remarks

NONE

Example

```
objStreaming.Is4VGAMode = TRUE;
```

See Also

7 New Events

This section lists new implement events.

New Events

Name	*Description*
OnPlayAudioFileComplete()	Event generated when the audio file play complete
OnMouseMove()	Event generated when the mouse move on the media area in the media window
OnAsyncURLCommandDone()	Event generated when the URL command send
OnTimeZoneEx()	Event generated when set Time Zone
OnTimeCodeEx()	Event generated when set Time code
OnMDSettingEx()	Event generated when receive motion setting
OnPIREventStart()	Event generated when PIR event start
OnPIREventEnd()	Event generated when PIR event end
RequestSetIOnly()	Event generated when request decode I only
OnDIEventEx()	Event generated when the DI is triggered
OnNewImage()	Event generated when new image
OnAutoReconnect	Event generated when auto-reconnect

OnPlayAudioFileComplete()

Description

Event generated when the audio file play complete.

Syntax

```
OnPlayAudioFileComplete( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnPlayAudioFileComplete(nID)" LANGUAGE="Jscript">
    Alert( "Play Complete" );
</SCRIPT>
'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_PlayAudioFileComplete()
        MsgBox "Play Complete"
    End Function
</SCRIPT>
```

See Also

OnMouseMove()

Description

Event generated when the mouse move on the media area in the media window.

Syntax

```
OnMouseMove( LONG nID, LONG nX, LONG nY )
```

Parameters

Name	*Type*	*Description*
nID	Long	Return camera id
nX	Long	X position
nY	LONG	Y position

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnMouseMove(nID,nX,nY)" LANGUAGE="Jscript">
    Alert( "Move" );
</SCRIPT>
'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnMouseMove()
        MsgBox "Move"
    End Function
</SCRIPT>
```

See Also

OnAsyncURLCommandDone()

Description

Event generated when the URL command send.

Syntax

```
OnAsyncURLCommandDone( LONG nID, LONG nIndex, BSTR szResult )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id
nIndex	LONG	Command index
szResult	BSTR	Result

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT     FOR=Media     EVENT="OnAsyncURLCommandDone(nID,nIndex,szResult)"
LANGUAGE="Jscript">
    Alert( "Send success" );
</SCRIPT>
'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnAsyncURLComandDone()
        MsgBox "Send Success"
    End Function
</SCRIPT>
```

See Also

OnTimeZoneEx()

Description

Event generated when Time Zone updates.

Syntax

```
OnTimeZoneEx( LONG nID, LONG timeVal, LONG timeZone, LONG fixTimezone )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id
timeVal	LONG	Return camera time
timeZone	LONG	Return camera timezone
fixTimezone	LONG	Return camera fix timezone

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT   FOR=Media   EVENT=”OnTimeZoneEX(nID,timeval,timezone,fixtimezone)”
LANGUAGE=”Jscript”>
    Alert( “set Success” );
</SCRIPT>
‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnTimeZoneEx()
        MsgBox “set Success”
    End Function
</SCRIPT>
```

See Also

OnTimeCodeEx()

Description

Event generated when set Time code.

Syntax

```
OnTimeCodeEx( LONG nID, LONG nTimeCode, LONG nMSec )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id
nTimeCode	LONG	Return camera time
nMSec	LONG	Return camera time million second

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT          FOR=Media          EVENT=”OnTimecodeEx(nID,nTimeCode,nMSec)”
LANGUAGE=”Jscript”>
    Alert( “set Success” );
</SCRIPT>
‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnTimeCodeEx()
        MsgBox “set Success”
    End Function
</SCRIPT>
```

See Also

OnMDSettingEx()

Description

Event generated when receive motion setting.

Syntax

```
OnMDSettingEx( LONG nID,
               LONG nIndex,
               LONG nX,
               LONG nY,
               LONG nW,
               LONG nH,
               LONG nSens,
               LONG enable,
               LONG nTime )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id
nIndex	LONG	Motion region index
nX	LONG	X position
nY	LONG	Y position
nW	LONG	Motion region width
nH	LONG	Motion region height
nSens	LONG	Motion region sensitivety
enable	LONG	Enable or disable
nTime	LONG	Motion region timer

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT                                                         FOR=Media
EVENT=”OnMDSettingEx(nID,nIndex,nX,nY,nW,nH,nSens,enable,nTime)”
LANGUAGE=”Jscript”>
    Alert( “Motion Success” );
</SCRIPT>
‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnMDSettingEx()
        MsgBox “Motion settings Success”
    End Function
</SCRIPT>
```

See Also

OnPIREventStart()

Description

Event generated when PIR event start.

Syntax

```
OnPIREventStart( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnPIREventStart(nID)" LANGUAGE="Jscript">
    Alert( "PIR Start" );
</SCRIPT>
'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnPIREventStart()
        MsgBox "PIR Start"
    End Function
</SCRIPT>
```

See Also

OnPIREventEnd()

Description

Event generated when PIR event end.

Syntax

```
OnPIREventEnd( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT FOR=Media EVENT="OnPIREventEnd(nID)" LANGUAGE="Jscript">
    Alert( "PIR End" );
</SCRIPT>
'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_OnPIREventEnd()
        MsgBox "PIR End"
    End Function
</SCRIPT>
```

See Also

RequestSetIOnly()

Description

Event generated when request decode I only.

Syntax

```
RequestSetIOnly( LONG nID, LONG bRequestSetIOnly )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>nID</i></td><td>Long</td><td>Return camera id</td></tr>
<tr><td><i>bRequestSetIOnly</i></td><td>Long</td><td>The set I only
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>No set I only</td></tr>
<tr><td><b>1</b></td><td>Set I only</td></tr>
</table>
</td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
'---- Java Script
<SCRIPT    FOR=Media    EVENT="    RequestSetIOnly(nID,bRequestSetIOnly)"
LANGUAGE="Jscript">
    Alert( "set I only" );
</SCRIPT>
'---- VB Script
<SCRIPT LANGUAGE="VBScript">
    Function Media_RequestsetIOnly()
        MsgBox "set I only"
    End Function
</SCRIPT>
```

See Also

OnDIEventEx()

Description

Event generated when the DI is triggered.

Syntax

```
OnDIEventEx( LONG nDIStatus )
```

Parameters

<table>
<tr><th><i>Name</i></th><th><i>Type</i></th><th><i>Description</i></th></tr>
<tr><td><i>nDIStatus</i></td><td>LONG</td><td>Return DI status
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>close</td></tr>
<tr><td><b>1</b></td><td>open</td></tr>
</table>
</td></tr>
</table>

Returns

NONE

Remarks

NONE

Example

```
‘---- Java Script
<SCRIPT FOR=Media EVENT=”OnDIEventEx(nDIStatus)” LANGUAGE=”Jscript”>
    Alert( “DI” );
</SCRIPT>
‘---- VB Script
<SCRIPT LANGUAGE=”VBScript”>
    Function Media_OnDIEventEx()
        MsgBox “DI”
    End Function
</SCRIPT>
```

See Also

OnNewImage()

Description

Event generated when new image.

Syntax

```
OnNewImage( LONG nID )
```

Parameters

Name	*Type*	*Description*
nID	LONG	Return camera id

Returns

NONE

Remarks

NONE

OnAutoReconnect()

Description

Event generated when auto-reconnect

Syntax

```
OnNewImage( LONG nID )
```

Parameters

Name	*Type*	*Description*
`nID`	`LONG`	Return camera id

Returns

NONE

Remarks

NONE