

Moxa VPort SDK PLUS

VPort SDK ActiveX Control

www.moxa.com/product

Fourth Edition, September 2008

MOXA®

© 2008 Moxa Inc. All rights reserved.
Reproduction without permission is prohibited.

Moxa VPort SDK PLUS

VPort SDK ActiveX Control

The content described in this document is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Copyright © 2008 Moxa Inc.
All rights reserved.
Reproduction without permission is prohibited.

Trademarks

MOXA is a registered trademark of Moxa Inc.
All other trademarks or registered marks in this manual belong to their respective manufacturers.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document “as is,” without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas:

Toll-free: 1-888-669-2872
Tel: +1-714-528-6777
Fax: +1-714-528-6778

Moxa China (Shanghai office):

Toll-free: 800-820-5036
Tel: +86-21-5258-9955
Fax: +86-10-6872-3958

Moxa Europe:

Tel: +49-89-3 70 03 99-0
Fax: +49-89-3 70 03 99-99

Moxa Asia-Pacific:

Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

Table of Contents

Chapter 1	Overview	1-1
	Introduction.....	1-2
	Getting Started with VPort SDK ActiveX Control.....	1-2
	Installing VPort SDK ActiveX Control	1-2
	Using VPort SDK ActiveX Control on the Web	1-2
	File Structure	1-3
	Installed Sample Files and Tools	1-3
	Release Notes.....	1-3
	Version 2.5	1-3
	Version 2.1	1-4
	Version 2.0	1-4
	Version 1.0	1-4
Chapter 2	Programmer's Guide.....	2-1
	Adding VPort SDK ActiveX Control to Your Project.....	2-2
	Application Sample Code	2-2
Chapter 3	Application Program Interface Reference	3-1
	Enumeration.....	3-2
	EConnectStatus.....	3-2
	EConnProtocol.....	3-2
	ERecordedStatus.....	3-3
	Properties	3-3
	AutoReconnect	3-3
	CHIndex	3-3
	ConnectionTimeout	3-3
	ConnectStatus	3-4
	DecodeAV	3-4
	DeInterlace	3-4
	Display.....	3-4
	EnableAudioControl.....	3-4
	ForceGDI.....	3-5
	GetStreamType.....	3-5
	HttpPort	3-5
	IgnoreBorder.....	3-5
	IgnoreCaption	3-6
	InterfaceAddress	3-6
	MagicCode	3-6
	MediaType.....	3-6
	MotionEditMode	3-7
	NotifyNewPacket.....	3-7
	NotifyVideoData	3-7
	Password.....	3-7
	RecordingManualNaming.....	3-8
	RecordMaxFileSize	3-8
	RecordMaxFileTimeLength	3-8
	RecordStatus.....	3-9
	RecordVideoPath.....	3-9
	ServerIP	3-9
	ServerModelType	3-9

ShowMDWindow	3-10
StreamingType	3-10
Username	3-10
VideoFrameRate	3-10
VideoProtocol	3-10
VoiceMute	3-11
VoiceVolume	3-11
Methods	3-11
AudioPost	3-11
Connect	3-11
Disconnect	3-12
GetClientReg	3-12
GetPresetName	3-12
GetSnapshotData	3-12
HttpCommand	3-13
PreviewFullScreen	3-13
SavePresetPosition	3-13
SaveSnapshot	3-14
SetClientReg	3-14
SetMulticastInfo	3-14
SendCameraCommand	3-15
StartRecording	3-15
StopRecording	3-15
Events	3-16
OnConnStatus	3-16
OnDIOStatus	3-16
OnDrop	3-16
OnMDAlert	3-17
OnMouseClicked	3-17
OnMouseDownClick	3-17
OnMouseDownClick	3-18
OnMouseMove	3-18
OnMouseUpClick	3-18
OnNewFrame	3-19
OnNotifySnapshotData	3-19
OnRMouseDownClick	3-20
OnRMouseDown	3-20
OnRMouseUp	3-20
OnRecordStatus	3-21
OnRequestFileName	3-21
OnResolutionChanged	3-21
OnVideoFrameRate	3-21

This document describes the properties and methods supported by **VPort SDK ActiveX Control**.

VPort SDK ActiveX Control supports Moxa's VPort 25 and products from the VPort 250 Series and VPort 350 Series.

The following topics are covered in this chapter:

❑ **Introduction**

- Getting Started with VPort SDK ActiveX Control
- Installing VPort SDK ActiveX Control
- Using VPort SDK ActiveX Control on the Web
- File Structure
- Installed Sample Files and Tools

❑ **Release Notes**

- Version 2.5
- Version 2.1
- Version 2.0
- Version 1.0

Introduction

Getting Started with VPort SDK ActiveX Control

The main function of VPort SDK ActiveX Control is to provide a rapid development of applications that can be used for monitoring, PTZ control, recording, and updating the configuration of the VPort. In addition, AVI conversion is also provided, and various development tools are supported, including Microsoft Visual Basic, Microsoft Visual C++, and script languages such as VBScript and JavaScript.

Installing VPort SDK ActiveX Control

Follow these steps to install VPort SDK ActiveX Control on your PC:

1. Download the installation file named **VPortSDKPLUS_ActiveX.exe**.
2. Double click the setup file and follow the onscreen instructions.
3. The default path for the software is **C:\Program Files\Moxa\VPortSDKPLUS_ActiveX**.

Using VPort SDK ActiveX Control on the Web

You can get a video/audio stream from your video servers by using the controls on your web pages. To access VPort SDK ActiveX Control services, use the HTML<OBJECT> tag and specify CLSID for the control. In addition, use the <PARAM> tag to assign the **ServerIP** property for the server to which you want to connect. In addition, assign a **username** and **password** to set the username and password for logging into the video server. The following example illustrates how to display video from IP address 192.168.0.100 in your web browser.

```
<OBJECT ID=" VPortsdkCtrl" WIDTH=460 HEIGHT=526
CLASSID="CLSID: 064A0198-F3AB-478A-8C04-EE647284D9AE"
CODEBASE=" VPortSDK.cab#version=1.0.0.0">
<PARAM NAME="ServerIP" VALUE="192.168.0.100">
<PARAM NAME="username" VALUE="admin">
<PARAM NAME="password" VAULE="admin">
<PARAM NAME="VideoProtocol" VALUE="1">
</OBJECT>
```

Please refer to the file **HTMLSampleCode.htm** in the **cab** subdirectory in the installation directory for more details. Use your own settings for the following items:

- **server.domain.hostname**: Use the IP or domain name + hostname.
- **plugin**: to the actual file path of the plug-in component.
- **username** and **password**: Set the username and password to log in to the video server. The defaults are username = admin, and password = admin.

File Structure

FILE	DESCRIPTION
doc\VPortSDK_ActiveX_Control.pdf	This is the manual file.
sample\VB	Sample code for VB
sample\VC6_Sample Code	Sample code for VC++ 6.0
sample\Csharp_Sample Code	Sample code for VS2005 C#
sample\VB.NET 2005_Sample Code	Sample code for VB.NET 2005
sample\VC++.NET 2005_Sample Code	Sample code for VC++.NET 2005

Installed Sample Files and Tools

FILE	DESCRIPTION
Cab\HTMLSampleCode.html	Sample code to connect to VPort 351
utility\ConvertPEStoAVI.exe	The tool to convert PES files to AVI files
utility\UPCOM.exe	Use this tool to install the VPort Real COM driver

Release Notes

Version 2.5

System Requirements

- Software: Windows 2000/XP/Vista(x86)

Functions that were Added or Modified

- Added: 4-channel function
- Modified: Connect issue in Windows Vista
- Added: CHIndex property
- Added: InterfaceAddress property
- Added: StreamingType property
- Added: AudioPost method
- Added: SetMulticastInfo method
- Modified: SaveSnapshot, which allows users to decide if they want to use the default file name.
- Added: GetSnapshotData method
- Added: NotifySnapshotData event
- Modified: The nameing rule in VC6
- Modified: SendCameraCommand add up-left, up-right, down-left, down-right directions and title stop, pan stop, zoom stop and stop command

Version 2.1

Added/Enhanced Features

- OnMouseDown event was added
- OnMouseMove event was added
- OnMouseUp event was added
- OnRMouseDown event was added
- OnRMouseUp event was added

Fixed Bugs

- Fixed the problems with reconnect
- Fixed the problems with dual monitor
- Solved the problem with system memory leaks

Version 2.0

Added / Enhanced Features

- RecordStatus property was added
- NotifyVideoData property was added
- NotifyNewPacket property was added
- OnMouseUp event was added
- OnMouseDown event was added
- Click event was added
- Modified the SaveSnapshot, which allows users to assign a name of their own choosing

Fixed Bugs

- Fixed the problems with multi-connect
- Fixed the problems with multi-recording
- Improved the DeInterlace effect
- Solved the problem with system blocks

Version 1.0

System Requirements

- Software: Windows 2000/XP

Features

- Supports events
- Supports two types of recording: PES or AVI recording
- Can control the display of received video data
- Can control the de-interlacing of decoded images
- Supports reconnection when the connection fails
- Supports PTZ camera control
- Supports saving snapshot images

2

Programmer's Guide

In this chapter, we discuss how to add SDK ActiveX Control to your project, and describe the various sample programs included with the VPort product.

The following topics are discussed in this chapter:

- ❑ **Adding VPort SDK ActiveX Control to Your Project**
- ❑ **Application Sample Code**

Adding VPort SDK ActiveX Control to Your Project

VC++ 6.0

Follow these instructions to add the SDK ActiveX Control to your project's toolbox:

1. From the **Project** menu, select **Add To Project/Components and Controls**. The Component and controls Gallery window will appear.
2. Open the **Registered VPortSDK Controls** folder, and choose **VPortSDK Control class** from the list.
3. Click the **Insert** button to close the window, and then click **OK** in the confirm window. Next, close the Components and Controls dialog. The VPortSDKCtrl object will now appear in the toolbox.
4. At this point, drag the VPortSDKCtrl object into the window.
5. The wrapped class VPortSDK is also available in your project workspace. The class can be used to access the control properties and methods.

VB 6.0

Follow these instructions to add the SDK ActiveX Control to the project:

1. Place the cursor over the toolbox and then click the right mouse button.
2. Select "Component" from the popup menu.
3. Select "VPortSDK ActiveX Control module"
4. Close "Component" window.
5. The VPort SDK Control icon will appear in the ToolBox.

Application Sample Code

Sample code for several examples is shipped with the product. The examples illustrate the following functions.

VB_Sample Code

This sample code is written in VB6.0, and demonstrates the following:

- Live streaming
- Calculation of frame rate, DIO status, and frame size
- Motion edit mode
- Miscellaneous options settings
- PES and AVI recording
- HTTP command function
- Saving snapshot pictures
- PTZ camera control

VC6_Sample Code

This sample code is written in VC++ 6.0, and demonstrates the following:

- Live streaming
- Calculation of frame rate, DIO status, and frame size
- Motion edit mode
- Miscellaneous options settings
- PES and AVI recording
- HTTP command function
- Saving snapshot pictures
- PTZ camera control

VB.NET 2005_Sample Code

This sample code is written in VB.NET 2005, and demonstrates the following:

- Live streaming
- Calculation of frame rate, DIO status, frame size
- Motion edit mode
- Miscellaneous options settings
- PES and AVI recording
- HTTP command function
- Saving snapshot pictures
- PTZ camera control

VC++.NET 2005_Sample Code

This sample code is written in VC++.NET 2005, and demonstrates the following:

- Live streaming
- Calculation of frame rate, DIO status, and frame size
- Motion edit mode
- Miscellaneous options settings
- PES and AVI recording
- HTTP command function
- Saving snapshot pictures
- PTZ camera control

Csharp_Sample Code

This sample code is written in VC#2005, and demonstrates the following:

- Live streaming
- Calculation of frame rate, DIO status, and frame size
- Motion edit mode
- Miscellaneous options settings
- PES and AVI recording
- HTTP command function
- Saving snapshot pictures
- PTZ camera control

Application Program Interface Reference

In this chapter, we describe the various functions that you can use with your applications.

The following topics are included in this chapter:

- ❑ **Enumeration**
 - EConnectStatus
 - EConnProtocol
 - ERecordedStatus
- ❑ **Properties**
 - AutoReconnect
 - CHIndex
 - ConnectionTimeout
 - ConnectStatus
 - DecodeAV
 - DeInterlace
 - Display
 - EnableAudioControl
 - ForceGDI
 - GetStreamType
 - HttpPort
 - IgnoreBorder
 - IgnoreCaption
 - InterfaceAddress
 - MagicCode
 - MediaType
 - MotionEditMode
 - NotifyNewPacket
 - NotifyVideoData
 - Password
 - RecordingManualNaming
 - RecordMaxFileSize
 - RecordMaxFileTimeLength
 - RecordStatus
 - RecordVideoPath
 - ServerIP
 - ServerModelType
 - ShowMDWindow
 - StreamingType
 - Username
- ❑ **Methods**
 - AudioPost
 - Connect
 - Disconnect
 - GetClientReg
 - GetPresetName
 - GetSnapshotData
 - HttpCommand
 - PreviewFullScreen
 - SavePresetPosition
 - SaveSnapshot
 - SetMulticastInfo
 - SetClientReg
 - SendCameraCommand
 - StartRecording
 - StopRecording
- ❑ **Events**
 - OnConnStatus
 - OnDrop
 - OnDIOStatus
 - OnMDAlert
 - OnMouseDownClick
 - OnMouseDownClick
 - OnMouseMove
 - OnMouseUpClick
 - OnNewFrame
 - OnNotifySnapshotData
 - OnRMouseDownClick
 - OnRMouseDown
 - OnRMouseUp
 - OnMouseClicked
 - OnRecordStatus
 - OnRequestFileName
 - OnResolutionChanged
 - OnVideoFrameRate
- VideoFrameRate
- VideoProtocol
- VoiceMute
- VoiceVolume

Enumeration

The enumerations in this section are available for VC and VB. If you need to pass a value of the following enumerations as parameters, please use the corresponding value.

EConnectStatus

List Members

Name	Value	Description
eConnecting	0	The control is now connecting to the remote server.
eConnected	1	The connection is already connected.
eDisconnect	2	The connection is now disconnecting.
eConnectFail	3	Failed to connect to the remote server.

Description

This enumeration is used with the **ConnectStatus** property.

EConnProtocol

List member

Name	Value	Description
eProtNone	0	The control is currently not connecting to Video Server.
eProtUDP	1	Uses UDP as the connecting protocol.
eProtTCP	2	Uses TCP as the connecting protocol.
eProtHTTP	3	Uses HTTP as the connecting protocol. HTTP protocol does not support audio.
eProtMulticast	4	Uses multicast to receive the streaming data. This is only valid for RTSP Models, and the firmware must support it.
ePortMulticastPush	5	Uses continuous multicast push as the connection protocol. This is only valid for 4-channel models.

Description

This enumeration is used with the set/get audio or video protocol.

ERecordStatus

List Member

Name	Value	Description
eNone	0	The control is currently not recording.
eRecording	1	The control is recording right now.
eStatusDiskfull	2	Disk is full when recording. The recording will be stopped when this status is stopped.
eStatusRecordStart	3	Sent to users when the location starts to record. This is often used with event recording to notify users that a new event has occurred, and that recording has started.
eStatusRecordStop	4	Sent to users when recording stops.

Description

This enumeration is used to specify the recording status.

Properties

AutoReconnect

Decide whether the control should try to reconnect to the server if it finds the connection broken.

Type

Boolean

True means to reconnect automatically and false means not to reconnect automatically.

Attribute

R/W

CHIndex

Set or get the current video channel index. This property is meaningful for 4-channel products.

Type

long

The video channel index

Attribute

R/W

ConnectionTimeout

Set or get the timeout value used when connected to the server. The default value is 5 seconds.

Type

Long

The value is in milliseconds.

Attribute

R/W

ConnectStatus

Retrieve the connection status. The status will be maintained during operation.

Type

EConnectStatus

This is the status for the connection.

Attribute

R/O

DecodeAV

Set the flag that decides if the audio and video data should be decoded after received.

Type

Boolean

True means to decode the audio and video data, and false means not to decode it.

Attribute

W/O

DeInterlace

Smooth the frame image when decoding the video stream.

Type

Boolean

True means to smooth the decode image, and false means not to smooth it.

Attribute

R/W

Display

Set or get the switch to turn on or turn off the display of the graph on screen.

Type

Boolean

True means to show the graph. False means to disable the display.

Attribute

R/W

EnableAudioControl

Show/hide the microphone controller options.

Type

Boolean

True means to show the microphone controller. False means to disable the microphone controller.

Attribute

R/W

ForceGDI

Decides if the control shows video in GDI mode, regardless of if the card supports DirectDraw.

Type

Boolean

Set this value to True to force video to be displayed in GDI mode. Default value is False.

Attribute

R/W

GetStreamType

Get the current stream type of the video server.

Type

Long

A return value of 1 means the stream is MPEG4, and 2 means Motion JPEG.

Attribute

R/O

HttpPort

The user should set the port number (usually 80) in VPortSDK Control to allow communication with the video server.

Type

Long

This is the port number. Default value is 80. If you change the HTTP port used by the Video Server, you should also update this value.

Attribute

R/W

IgnoreBorder

Set to true if the control should display the border when showing video. The border is a gray line with 5-pixel width enclosing the video. When the value is true, the border will be ignored. The default value is false.

Type

Boolean

Show or hide the border.

Attribute

W/O

IgnoreCaption

Set to true if the control should display the caption when showing video. The caption is the 25 pixels high text line above the video. When the value is true, the caption will be ignored. The default value is false.

Type

Boolean

Show or hide the caption.

Attribute

W/O

Remarks

When in Motion editing mode, the caption cannot be ignored.

InterfaceAddress

Set or get the local interface address for 4-channel continuous multicast push connections.

Type

String

The local interface address

Attribute

R/W

Remark

The returned string contains the local interface address that connected with “;”

MagicCode

Get or set the remote ID string. The remote ID string is an encrypted string that contains the user name and password used to connect to the remote server.

Type

String

This is the string content.

Attribute

R/W

MediaType

Set the media type for connection.

Type

EMediaType

This is the new media type used by this control. The value is one of the values in **EMediaType**.

Attribute

W/O

MotionEditMode

Show/hide the Video Motion Detection options.

Type

Boolean

Use True to show the Motion Detection options. False disables the options.

Attribute

R/W

NotifyNewPacket

Determine if the control should send the control owner each packet received from the network.

Type

Boolean

True means to notify for each packet. False means not to do so.

Attribute

R/W

Remark

The default value is False. Note this property is different from **NotifyVideoData**. This packet notifies un-decoded data, but **NotifyVideoData** notifies for decoded video.

NotifyVideoData

Determine if the control should send the control owner new video decoded data by event.

Type

Boolean

True means to notify for each piece. False means not to do so.

Attribute

R/W

Remark

Notifying too frequently could cause a performance penalty. The default value is False.

Password

Get or set the password used for web page authentication.

Type

String

Attribute

W/O

RecordingManualNaming

This property indicates if the record file name is given by the user or generated by the control itself. If this property is set to true, the control will fire the event **OnRequestFileName** when a new AVI file name is needed. The application can give the file name by setting the property

RecordFileName.

Type

Boolean

True if the record file name is given by the application. False if the name is generated by the control.

Attribute

R/W

Remarks

If you set "RecordingManualNameing" to true but then do not implement the event (i.e., OnRequestFileName → To set up the file name), then the VPortSDK Control will create a filename automatically. This is done to ensure that the VPort will continue to operate normally.

RecordMaxFileSize

This property is used to set the maximum record file size.

Type

Long

If your operating system (e.g., 2000/XP) uses the NTFS file system, then there is essentially no upper limit to the size of AVI files. If the FAT32 file system is used, then AVI files are limited to 4 GB.

Attribute

R/W

Remarks

The default minimum size of the file is 1 MB. The size of the hard disk must be at least 1 GB, or it will not start recording.

RecordMaxFileTimeLength

This property is used to set the maximum time associated with each AVI file. For example, if the recording time is 10 minutes (600 seconds), then setting the RecordMaxFileTimeLength to 60 seconds will cause the recording to be saved in 10 separate AVI files.

Type

Long

The minimum allowed value is 5 seconds. The maximum is 3600 seconds (1 hour), although you will seldom use such a large value. Setting this value to 0 disables the time limitation.

Attribute

R/W

RecordStatus

Retrieve the control's recording status. Note that the control recording status is automatically reset to false after the connection is closed.

Type

ERecordedStatus

This is the status of the recording.

Attribute

R/O

RecordVideoPath

This property is used for applications to specify the name of the next AVI file generated. The name should include the full path name.

Type

String

This is the file name set by the application.

Attribute

R/W

Remarks

The default record file path is "C:\".

ServerIP

Get or set the remote IP address of the Video Server.

Type

String

The format should be dotted IP: 'a.b.c.d'

Attribute

R/W

ServerModelType

Get the server model type of the target to be connected.

Type

String

This is the type of the server.

Attribute

R/O

ShowMDWindow

Determines if the control will show red frames.

Type

Boolean

True means to draw the red frame, and false means not to draw it.

Attribute

R/W

StreamingType

Get or set the video streaming type.

Type

long

1 means the MPEG4 type, and 2 means MJPEG.

Attribute

R/W

Remark

Just for 4-channel multiple codec models.

Username

Set or get the user name for web authentication.

Type

String

Attribute

R/W

VideoFrameRate

Retrieve the frame rate of the current connection. This is only applicable for those connections that contain video data.

Type

Long

This is the frame rate in FPS (frames per second).

Attribute

R/O

VideoProtocol

Set the connection protocol that is used when connecting to the server.

Type

EConnProtocol

The available protocol is listed in **EConnProtocol**.

Attribute

W/O

VoiceMute

Turn on or off the audio when playing.

Type

Boolean

True means to turn on the audio, and false means to turn off the video.

Attribute

R/W

VoiceVolume

Get or set the audio volume used when playing.

Type

Long

The value range is 0-100. This is where 0 is equivalent to mute and 100 is the loudest level.

Attribute

R/W

Methods

AudioPost

Post audio source to video server.

Syntax

```
BOOL AudioPost ( BOOL bEnable );
```

Return Value

TRUE: OK

FALSE: ERROR

Parameters

bEnable

[in] Enable the audio post function

Connect

Connect to the Video Server to start the download.

Syntax

```
Long Connect ( );
```

Return Value

0: OK

< 0: ERROR

Parameters

None

Disconnect

Disconnect the Video Server to stop the download. This call is in blocking mode, so it will not be returned until the connection is truly disconnected.

Syntax

```
Long Disconnect ( );
```

Return Value

0: OK
< 0: ERROR

Parameters

None

GetClientReg

Reserved.

GetPresetName

Retrieve the list of names of the camera's preset positions.

Syntax

```
BSTR GetPresetName( );
```

Return Value

Return the preset point name.

Parameters

None

Remarks

The returned string contains the preset point names that connected with "#".

GetSnapshotData

It is used to set the decoded video data format.

Syntax

```
Void GetSnapshotData( long IPicFormat );
```

Return Value**Parameters**

IPicFormat
[in] The format of image.

Remark

After Setting this method, please get the decoded video data in NotifySnapshotData event.

HttpCommand

Send http command to server by either POST or GET.

Syntax

```
BSTR HttpCommand ( LPCTSTR lpzSendType,  
                   LPCTSTR lpzServerIP,  
                   LPCTSTR lpzURLString,  
                   BOOL bReturnVal );
```

Return Value

Return the server's data

Parameters

lpzSendType

[in] Is the command to be sent as POST or GET?

lpzServerIP

[in] This is the server's IP address.

lpzURLString

[in] This is the URL command to be sent to the server.

bReturnVal

[in] Should the control try to get the returned page data? If this argument is set to False, the argument following it will be ignored.

PreviewFullScreen

Reserved.

SavePresetPosition

Save the current position of the camera as a preset setting in the server.

Syntax

```
void SavePresetPosition ( LPCTSTR lpzPresetName,  
                          Short sPresetIndex );
```

Return Value

Parameters

lpzPresetName

[in] This is the name of the position to be set.

sPresetIndex

[in] This is the index of the position to be set.

SaveSnapshot

Save the current image frame to a specified file.

Syntax

```
void SaveSnapshot ( LPCTSTR lpszSavePath,  
                   LPCTSTR lpszSaveFileName  
                   BOOL bAuto );
```

Return Value

Parameters

lpszSavePath

[in] This parameter sets the snapshot's save path.

lpszSaveFileName

[in] This parameter set the snapshot's file name.

bAuto

[in] This parameter enables/disables the default file name.

SetClientReg

Reserved.

SetMulticastInfo

Sets parameters of continuous multicast push for 4-channel video servers.

Syntax

```
void SetMulticastInfo ( LPCTSTR lpstIP,  
                       LPCTSTR lpstVideoPort,  
                       LPCTSTR IpstAudioPort );
```

Return Value

Parameters

lpstIP

[in] The multicast group address.

lpstVideoPort

[in] The multicast video port.

IpstAudioPort

[in] The multicast audio port.

SendCameraCommand

Send camera control commands to the server.

Syntax

```
long SendCameraCommand ( LPCTSTR lpzStrCommand,  
Short sCameraParam );
```

Return Value

0: OK
< 0: ERROR

Parameters

lpzStrCommand

[in] This is the command supported by the server at that time: right(right), left(left), pan stop(pstop), up(up), down(down), title stop(tstop), zoom-in(zin), zoom-out(zout), zoom stop(zstop), go-preset(gopreset), up-left(upleft), up-right(upright), down-left(downleft), down-right(downright), stop(stop).

sCameraParam

[in] Set the speed at which the PTZ camera moves.

StartRecording

Start recording the video file. This method can be used to record the video file in PES or AVI format.

Syntax

```
long StartRecording ( Short slRecordFormat );
```

Return Value

0: OK
< 0: ERROR

Parameters

sRecordFormat

[in] Indicate the record type: 0 for PES and 1 for AVI.

StopRecording

Stop the video recording.

Syntax

```
long StopRecording ( Short sRecordFormat );
```

Return Value

0: OK
< 0: ERROR

Parameters

sRecordFormat

[in] Indicates the record type: 0 for PES and 1 for AVI.

Events

This control supports connection points. The control owner can receive certain events when certain conditions are true. To receive these events, VC users should implement the event-sinking interface. Readers can find an example of how to implement the event-sinking interface using MFC in the sample codes. For those that don't use MFC, search the Internet for the ATL implementation of a sinking target. VB users can implement the events easily by clicking on the **Procedures/Events Box** to insert the events.

OnConnStatus

The control fires this event whenever the connection status is changed.

Syntax

```
void OnConnStatus ( Long IConnectionStatus );
```

Return Value

Parameters

IConnectionStatus

[in] Indicates the status code for this notification. The status code is defined in **EConnectStatus**.

OnDIOStatus

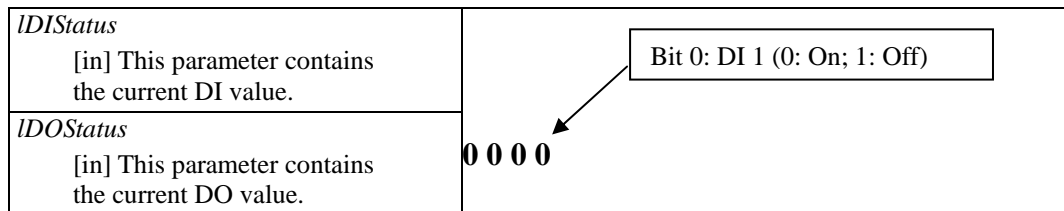
The control fires this event whenever it gets the DI / DO status from the video stream.

Syntax

```
void OnDIOStatus( Long IDIStatus, Long IDOStatus );
```

Return Value

Parameters



OnDrop

Reserved.

OnMDAlert

The control fires this event whenever it gets a motion detection alert from the video stream.

Syntax

```
void OnMDAlert ( Long lMotionID,  
                Long lPercentage );
```

Return Value

Parameters

lMotionID

[in] This parameter indicates the IDs of the different MD windows.

lPercentage

[in] This parameter indicates the variation of the different MD window.

OnMouseClicked

The control fires this event whenever the user clicks on the control by using mouse pointer.

Syntax

```
void OnMouseClicked ( Long lX ,  
                     Long lY );
```

Return Value

Parameters

lX

[in] The x coordinate related to upper-left corner of the control where user clicked mouse.

lY

[in] The y coordinate related to upper-left corner of the control where user clicked mouse.

OnMouseDownClick

The control fires this event whenever the user clicks on the control with the mouse pointer.

Syntax

```
void OnMouseDownClick ( Long lX ,  
                       Long lY );
```

Return Value

Parameters

lX

[in] The x coordinate related to upper-left corner of the control where user double clicked mouse.

lY

[in] The y coordinate related to upper-left corner of the control where user double clicked mouse.

OnMouseDownClick

The control fires this event whenever the user presses a mouse button.

Syntax

```
void OnMouseDownClick (    Long  IX ,  
                          Long  IY );
```

Return Value

Parameters

IX

[in] The x coordinate related to upper-left corner of the control where user presses the mouse button.

IY

[in] The y coordinate related to upper-left corner of the control where user presses the mouse button.

OnMouseMove

The control fires this event when the user moves the mouse pointer.

Syntax

```
void OnMouseMove (    Long  IX ,  
                    Long  IY );
```

Return Value

Parameters

IX

[in] The x coordinate related to upper-left corner of the control where user moves the mouse pointer.

IY

[in] The y coordinate related to upper-left corner of the control where user moves the mouse pointer.

OnMouseUpClick

The control fires this event whenever the user releases a mouse button.

Syntax

```
void OnMouseUpClick (    Long  IX ,  
                        Long  IY );
```

Return Value

Parameters

IX

[in] The x coordinate related to upper-left corner of the control where user releases the mouse button.

IY

[in] The y coordinate related to upper-left corner of the control where user releases the mouse button.

OnNewFrame

The control fires this event whenever a new image frame is received from the server. This event is not always fired. When the NotifyVideoData property is set to true, it will be set to notify that new video data will be transmitted.

Syntax

```
void OnNewFrame ( Long lFrameType,  
                  Long lFrameSize,  
                  Bool bVideoLoss,  
                  Variant *pvFrameBuffer );
```

Return Value

Parameters

lFrameType

[in] Indicates the new frame types.

lFrameSize

[in] Indicates the new frame size.

bVideoLoss

[in] True means signal on, and False means signal loss.

pvFrameBuffer

[in] This parameter contains the new frame data.

OnNotifySnapshotData

The control fires this event whenever a image data is decoded. This event is not always fired. When the GetSnapshotData method is set, it will be set to notify that decoded video data will be transmitted.

Syntax

```
void OnNotifySnapshotData( Long lPicFormat,  
                           Long lWidth,  
                           Long lHeight,  
                           Long lDataSize,  
                           Variant *pvData );
```

Return Value

Parameters

lPicFormat

[out] the video data format

lWidth

[out] the video data width

lHeight

[out] the video height

lDataSize

[out] the video data size

pvData

[out] the video data buffer

OnRMouseDownClick

The control fires this event when the user double-clicks the right mouse button.

Syntax

```
void OnRMouseDownClick(    Long  IX ,  
                          Long  IY );
```

Return Value

Parameters

IX

[in] The x coordinate related to upper-left corner of the control where the user double-clicks the right mouse button.

IY

[in] The y coordinate related to upper-left corner of the control where the user double-clicks the right mouse button.

OnRMouseDown

The control fires this event when the user presses the right mouse button.

Syntax

```
void OnRMouseDown (    Long  IX ,  
                     Long  IY );
```

Return Value

Parameters

IX

[in] The x coordinate related to upper-left corner of the control where the user presses the right mouse button.

IY

[in] The y coordinate related to upper-left corner of the control where the user presses the right mouse button.

OnRMouseUp

The control fires this event when the user releases the right mouse button.

Syntax

```
void OnRMouseUp (    Long  IX ,  
                  Long  IY );
```

Return Value

Parameters

IX

[in] The x coordinate related to upper-left corner of the control where the user releases the right mouse button.

IY

[in] The y coordinate related to upper-left corner of the control where the user releases the right mouse button.

OnRecordStatus

The control fires this event whenever an error happens during recording.

Syntax

```
void OnRecordStatus ( Long lStatus );
```

Return Value

Parameters

lStatus

[in] Indicates the status code for this notification. The status code is defined in **ERecordedStatus**. A value of 2 means the disk is full, and no space is left for recording. Note that the disk full event is handled by stopping the recording, or this event will occur each time network packets arrive.

OnRequestFileName

The control fires this event when a new record file name is needed.

Syntax

```
void OnRequestFileName ( Variant *pvTime );
```

Return Value

Parameters

pvTime

[in] Indicates the application's default file name.

OnResolutionChanged

The control fires the event when the video image size has been changed.

Syntax

```
void OnResolutionChanged ( Long lWidth,  
                           Long lHeight );
```

Return Value

Parameters

lWidth

[in] Indicates the width of the video frame image.

lHeight

[in] Indicates the height of the video frame image.

OnVideoFrameRate

Retrieve the frame rate of the current connection. This is only applicable for connections that transmit video data.

Syntax

```
void OnVideoFrameRate ( Long lFrameRate );
```

Return Value

Parameters

lFrameRate

[in] This is the frame rate in fps (frame per second).