Vitamin Control

Version 2.1.0.0

2006/11/1

TABLE of CONTENTS

CHAPTER 1 OVERVIEW 8

1.1 INTRODUCTION 9

Getting Started with Vitamin Control 9

Installing the ActiveX control 9

Using the Vitamin Control in web pages 9

Turn off the error message box 10

Record via the control 10

File Structure 11

Installed sample files & tool 11

1.2 RELEASE NOTE 12

Version 2.1.0.0 12

Version 2.0.0.0 15

Version 1.0.0.4 18

CHAPTER 2 PROGRAMMER'S GUIDE 19

2.1 USING VITAMIN CONTROL 20

Adding ActiveX Control to your Project 20

Create a Member Variable for VitaminCtrl 21

Set Visual Server contact information to Get Live Video/Audio streams 21

Control PTZ camera attached to server 22

Save current image to a file 23

Get current image data in your program 24

How to load Vitamin control dynamically 24

2.2 APPLICATION SAMPLE CODE 28

Vbtest 28

Vctest 28

CSharptest 29

dynaloadcsharp 30

dynaloadvb 30

dynaloadvc 30

vcstring 30

testcsharpmdi 30

CHAPTER 3 APPLICATION PROGRAM INTERFACE REFERENCE 31

3.1 ENUMERATION 32
EAudioCodecType *33*
EAVIRecordStatus *34*
EControlStatus *35*
EClickEventHandler *36*
EConnectionType *37*
EConnProtocol *38*
EControlButtonState *39*
EControlType *40*
ECoordinateType *41*
EDBRecordEventType *42*
EDBStatusCode *44*
EDisplayTimeFormat *45*
EMediaType *46*
EPanelBtnStyle *47*
EPictureFormat *48*
EPTZEnableFlag *49*
ERegistryRoot *50*
EServerConfig *51*
EServerModelType *52*
ESpeedType *53*
EStreamingOption *54*
ETalkBtnStyle *55*
EVideoCodecType *56*
EVideoQuality2K *57*
EVideoSize2K *58*
3.2 PROPERTIES 59
AudioBitRate *60*
AutoAVISettings *61*
AutoReconnect *62*
AutoServerModelType *63*
AVIFilePathName *64*
AVIManualNaming *65*
AVIMaxFileSize *66*
AVIMaxFileTimeLength *67*
AVIPath *68*
AVIStatus *69*
AVIVideoFrameRate *70*

AVIVideoHeight *71*
AVIVideoSizeByStream *72*
AVIVideoWidth *73*
BitmapFile *74*
CircularMode *75*
ClickEventHandler *76*
ControlID *77*
ControlPort *78*
ConnectionProtocol *79*
ConnectionTimeout *80*
ControlButtonOpts *81*
ControlStatus *82*
ControlType *83*
CurrentAudioCodecType *84*
CurrentAudioPort *85*
CurrentAudioProtocol *86*
CurrentControlCam *87*
CurrentMediaType *88*
CurrentProtocol *89*
CurrentVideoCodecType *90*
CurrentVideoPort *91*
DatabasePath *92*
DBHierarchy *93*
Deblocking *94*
DecodeAV *95*
DigitalInURL *96*
DigitalOutURL *97*
DigitalZoomEnabled *98*
DigitalZoomEnableChk *99*
DigitalZoomFactor *100*
DigitalZoomX *101*
DigitalZoomY *102*
Display *103*
DisplayMotionFrame *104*
DisplayPeriod *105*
DisplayTimeFormat *106*
DisplayErrorMsg *107*
DDrawOnePass *108*

DrawHwnd *109*
EnglishString *110*
EventTypes *111*
ForceGDI *112*
ForceNonYUV *113*
FrameRate *114*
GDIUseStretchBlt *115*
GetMaskEditParmUrl *116*
GetMDParmUrl *117*
HttpPort *118*
HWnd *119*
IgnoreBorder *120*
IgnoreCaption *121*
IndexSize *122*
IsRecording *123*
JpegQuality *124*
JpegSecsPerFrame *125*
JpegURL *126*
Language *127*
LeftTitleSpace *128*
Location *129*
MaxFileSize *130*
MaxLocationSize *131*
MDEditMode *132*
MediaType *133*
MediaRecord *134*
MicMute *135*
MicVolume *136*
NotifyImageFormat *137*
NotifyAudioPacket *138*
NotifyNewAudio *139*
NotifyVideoData *140*
NotifyVideoPacket *141*
PanelButtonStyle *142*
Password *143*
PlayMute *144*
PlayVolume *145*
PostEventTime *146*

PrebufferMemorySize *147*
PreEventTime *148*
PresetURL *149*
PtzURL *150*
RecallURL *151*
ReadSettingByParam *152*
ReadWriteTimeout *153*
ReconnectionWait *154*
RegkeyRoot *155*
RegSubKey *156*
RemoteIDStr *157*
RemoteIPAddr *158*
RemotePort *159*
RightTitleSpace *160*
ServerConfig *161*
ServerConfigEntry *162*
ServerConfigSection *163*
ServerModelType *164*
SetMaskEditParmUrl *165*
SetMDParmUrl *166*
Stretch *167*
StreamingOption *168*
TalkButtonStyle *169*
TextOnVideo *170*
TitleBarColor *171*
TitleTextColor *172*
UartURL *173*
Url *174*
UserDateFormat *175*
UserName *176*
VideoBitRate *177*
VideoQuality2K *178*
VideoSize2K *179*
WheelEventHandler *180*
3.3 METHODS 181
CloseConnect *182*
ChooseAVIAudioCompressor *183*
ChooseAVIVideoCompressor *184*

Connect *185*
Disconnect *186*
GetConnectionStatus *187*
GetDigitalIn *188*
GetPtzPresetPosition *189*
GetSnapshot *190*
GetUartData *191*
GetUartDataBinary *193*
HttpCommand *195*
InputMediaPacket *197*
InputMediaPacketX *198*
RecallPtzPosition *199*
RefreshServerConfig *200*
RepairDatabase *201*
RepairLocation *203*
RestoreControlHandle *205*
SavePresetPosition *206*
SaveSendMail *207*
SaveSnapshot *208*
SendCameraCommand *209*
SendCameraCommandMap *210*
SendCameraControlSpeed *211*
SendDigitalOut *212*
SendUartCommand *213*
SendUartCommandBinary *214*
SetBitmapHandle *215*
SetDatabasePath *216*
SetGivenLangInfo *217*
SetLangString *219*
SetLangStringHex *220*
SetLocation *222*
SetServerConfig *223*
SetServerDateTime *224*
StartAVIConversion *225*
StartMediaRecord *226*
StartMediaRecordEx *227*
StartPacketInput *229*
StopAVIConversion *230*

StopMediaRecord *231*
StopPacketInput *232*
UpdateServerConfig *233*
3.4 EVENTS 235
OnAVIStatus *236*
OnClick *237*
OnConnectionBroken *238*
OnConnectionOK *239*
OnDIDOAlert *240*
OnMDAlert *241*
OnNewAudioPiece *242*
OnNewPacket *243*
OnNewVideo *244*
OnRecordStatus *246*
OnRequestAVIFileName *247*
OnServerModelType *248*
3.5 ERROR CODE LIST 249

Chapter 1

Overview

1.1 Introduction

This document describes the properties and methods supported by the Vitamin ActiveX control. Vitamin ActiveX control, in this release, supports all series model products.

Getting Started with Vitamin Control

The main function of Vitamin Control is to provide a rapid development of the application that could be used to monitor, (PTZ or UART) control, record, and configuration update. It also provides decoded and compressed data access and AVI conversion. It supports various development tools such as Microsoft Visual Basic, Microsoft Visual C++ and Microsoft C#, and script languages such as VBScript and JavaScript.

Installing the ActiveX control

To install the Vitamin Control on your PC, just follow these steps:

1. Download or the installation file VitaminDecoder.zip and unpack it. If you get the distribution disc, just find the VitaminDecoder.exe under lib subdirectory.
2. Run the VitaminDecoder.exe.
3. The installed directory contains two subdirectories. The control itself is under lib and the cab-file and the sample html file using this cab-file is under cab subdirectory.

Using the Vitamin Control in web pages

You can get video/audio stream from VS server by using the control in your web pages. To access the Vitamin Control services, use HTML <OBJECT> tag and specify CLSID for the control. In addition, use the <PARAM> tag to assign property RemoteIPAddr for the server that you want to connect to. Also assign the UserName and Password to pass the authentication of server. And also decide the server model by assign ServerModelType to tell the control what kind of server you want to connect to. The following example illustrates how to get video from 192.168.0.100 in your web page, suppose it’s a 6000 series server:

```
<OBJECT ID="VitaminDecoder" WIDTH=362 HEIGHT=270
```

```
CLASSID="CLSID: 70EDCF63-CA7E-4812-8528-DA1EA2FD53B6"
CODEBASE="VitCtrl.cab#version=1,0,0,4">
<PARAM NAME="RemoteIPAddr" VALUE="192.168.0.100 ">
<PARAM NAME="UserName" VALUE="root ">
<PARAM NAME="Password" VALUE="123">
<PARAM NAME="ServerModelType" VALUE="2 ">
</OBJECT>
```

Note: Please refer to the sample.html under the cab subdirectory of installed directory for more detail. Replace the followings with your settings:

- **"server.domain.hostname" with the IP or domain name + hostname**
- **"plugin" for CODEBASE to the actual path on your web server.**
- **"username" and "password" with the user you open for the internet user, usually, this is the demo user in video server.**

The other samples miscsample_xx.html allows you to control more on the control by setting properties and calling functions of this control.

Turn off the error message box

If the control could not connect to server, there would be an error message box showing message. For users that does not need that message, please set the DisplayErrorMsg property to False.

Record via the control

The control supports two kinds of recording: one is proprietary format database recording. The other is AVI file base recording. For the database recording, users could use the MediaDBPlayback control to play the data back. For AVI recording, users could use standard tool to play the AVI file back. These two methods are for different purpose, the former use much less CPU power so that it is possible to run up to tens of controls at the same time. But the latter way consumes much CPU power to encode the video into selected codec format. So in a moderate PC, it might stress the CPU usage to 100% if two or more controls record at the same time. Please refer to the StartMediaRecord, StartMediaRecordEx function for database recording. Please refer to StartAVIConversion for AVI recording.

File Structure

FILE	DESCRIPTON
doc\ VitaminControl.doc	This manual
lib\VitaminDecoder.exe	The installation file for the control itself and the cab file that signs the control.
sample\csharptest	Sample code for C#
sample\vbtest	Sample code for VB
sample\vctest	Sample code for VC++ 6.0
sample\dynaloadvb	Sample code for VB to load control dynamically
sample\dynaloadcsharp	Sample code for C# to load control dynamically
sample\dynaloadvc	Sample code for VC++ 6.0 to load control dynamically
sample\vcstring	Sample code for VC++ 6.0 to retrieve English string table

The csharptest, vbtest and vctest are designed to be able to connect to every supported model. Other sample codes are just for one model. But it is easy to extend to apply for other models.

Installed sample files & tool

FILE	DESCRIPTON
cab\mistsample_7k.html	Sample code to connect to 7K
cab\mistsample_6k.html	Sample code to connect to 6K
cab\mistsample_3k.html	Sample code to connect to 3K
cab\mistsample_2k.html	Sample code to connect to 2K with 1 channel
cab\mistsample_2k4ch.html	Sample code to connect to 2K with 4 channels
cab\sample.html	Simple sample code to view video and play audio, The setting value must be changed before it could be used
utility\GenerateRemoteIDString.exe	The tool to generate encrypted RemoteIDStr

1.2 Release note

Version 2.1.0.0

- System Requirements
  Software:
  1. Windows 98/ME/2000/XP
- Added/Enhanced Features
  - InputPacket now could accept the packets got by application itself (not from DataBroker or other VitaminCtrl)
  - Add SetGivenLangInfo, SetLangString, SetLangStringHex, and EnglishString for multi-lingual application.
  - Add AVIFilePathName, AVIManualNaming, and OnRequestAVIFileName to let application could control the file name of the AVI file.
  - Add AVIMaxFileSize, and AVIMaxFileTimeLength to let application control the file size and time range of an AVI file.
  - Support multicast for 7000 servers more thoroughly.
  - Remove the client setting pages for all models. So the EControlType is modified.
- Fixed bugs
  - 3000 timeout is too small, fixed
  - When save snapshot, the return code is not reset, so AP would treat it as error though it is done successfully.
  - Digital zoom adjustment would cause crash, fixed
  - Mobile audio quality is improved by enlarge the internal buffer
  - Fix the hanging problem when server time change to future.
  - Audio focus is changed to global so that the audio sound is still playing when window focus changes

- The talk mechanism in half duplex mode has bug, fixed
- Time zone problem in server push video (mjpeg), in Nepal. Fixed
- Add limitation in Motion detection window name. The ‘&’ and ‘=’ are blocked.
- Snapshot for BMP return 0 height and width. Fixed
- Private mask edit for 2403 has a problem for window size not reflected correctly if Windows sets to “not change window content during drag”. Fixed.
- Zoom function for given PtzUrl (different from Url) has problem. Fixed
- For 7000 servers that do not provide Access Name in sysinfo.cgi would cause connection error. We do work around here.
- The maximum URL length is changed to 1023 characters.
- Some ATI card would suffer memory leak problem when using DirectDraw mode. We fix it by using different painting function.
- The control would not show video correctly when using in C# MDI (the control is created when form created, not dynamically). Fixed
- If the source graph is smaller than draw window size, and stretch is off, there would be problem for refreshing for DirectDraw One Pass mode. Fixed.
- If there are two or more controls. The stretch settings for these two controls are different (one stretch and one not), and the source is smaller than draw window. The non-stretch one would have incorrect background. Fixed
- Single Jpeg could not notify packet in previous version. Fixed

- Known bugs

Version 2.0.0.0

- System Requirements
  Software:
  2. Windows 98/ME/2000/XP
- Added/Enhanced Features
  - The bitmap shown when not connected could be jpeg now. But the width must be multiply of 16
  - RTSP port is now assigned by control port
  - Snapshot could now support all image formats (but jpeg would need some conversion power if the streaming format is not jpeg)
  - 7000 caption is now the same as 3000 is
  - Add custom command invocation for camera command. Use “custxxx” as the command for SendCameraCommand. Where xxx is the command index. 1-10 is the range.
  - Add iris control for SendCameraCommand, the syntax is irisauto, open, close.
  - Add DisplayMotionFrame to let user controls if the motion window should be displayed when motion triggered.
  - Add multicast protocol for RTSP. Note this protocol depends on server, if server does not support, the control will switch to UDP mode.
  - Add ForceNonYUV property to solve the green screen problem in certain display card.
  - Add support for VS2403, SetMaskEditParmUrl and GetMaskEditParmUrl are added. And the control now has one more control type for mask editing.
  - The StartAVIConversion now could be called in OnConnectionOK. Though we don’t recommend to use in such way.
  - Add IYUV and YV12 support for OnNewVideo and

snapshot
- Add TextOnVideo property
- Add DisplayPeriod support
- Add DBHierarchy property
- Add PrebufferMemorySize property
- Add AutoServerModelType property
- Add OnServerModelType event
- Use new database engine, the new database engine is more reliable for auto recovery and the performance is better. But it consumes more memory for pre-buffer.

- Fixed bugs
  - Fix the bug that 6000 client will not close socket in some cases
  - 6000 connection would suffer from packet loss if use UDP protocol even in LAN. Fixed
  - If the system does not support sound card and the control is set to the audio only mode, there would be problem. Fixed
  - Fix the RTSP mode crash bug when system load is heavy
  - If the bitmap is located in remote host and the control is closed before the bitmap retrieval done, the control will crash. Fixed
  - When in RTSP mode and the control switches to video only, the shown media type is incorrect and sometimes it will crash. Fixed
  - RTSP mode would have problem when switch to HTTP protocol. Fixed
  - Host name starts with digital number would not connected, fixed
  - 6000 server set to mute and client set to audio only would cause control crashing. Fixed.
  - OnNewVideo would be blocked after long run. Fixed
  - The 6000 server's talk-button does not behave well after control reconnected. Fixed
  - For model does not support click on image, click

on image and closing program will hang for 20 seconds. Fixed

- The Recall command is changed to GET instead of POST to work for all models.
- Fix AVI conversion for file's time length information.
- Support more audio codecs for AVI conversion.
-

- Known bugs

Version 1.0.0.4

- System Requirements
  Software:
  1. Windows 98/ME/2000/XP
- Features
  - Several handling selections for mouse click event.
  - Support reconnection when connection broken.
  - Controlling for display of received video data.
  - Controlling for de-blocking of decoded image.
  - Support several camera control functions.
  - Support reading data from /write data to UART on VS server.
  - Support events.
  - Support two types of recording: proprietary DB or AVI recording. In the former, it also support event recording and cycling recording feature.
  - User could choose to receive decoded or un-decoded data (both audio and video)
  - Support server settings read and update
  - Support full set of the server models

Chapter 2

PROGRAMMER'S GUIDE

2.1 Using Vitamin control

Adding ActiveX Control to your Project

VC++ 6.0

To add a control into project's toolbox

1. From the **Project** menu, select **Add To Project/Components and Controls**. The Component and Controls Gallery dialog shows up.
2. Open the **Registered ActiveX Controls** folder. Choose **VitaminCtrl class** in the list box.
3. Click **Insert** button to close the dialog, and click OK on the confirm dialog. Then close the Components and Controls dialog. The VitaminCtrl object will now appear in the toolbox.
4. Now you can drag the VitaminCtrl object into your dialog.
5. The wrapped class CVitaminCtrl is also available in your project workspace. The class could be used to access the control properties and method.

VB 6.0

To add a control into the project:

1. Please mouse content help (right button if the mouse button is not switched) button on ToolBox panel.
2. Choose "Component" item in the popup menu.
3. Locate and check VitaminDecoder 1.0 Type Library
4. Close Component dialog.
5. The Vitamin Control icon will appear in the ToolBox. You could use the control hereafter.

C

Too add a control into the project

1. Move mouse to the ToolBox panel and wait until the panel shows up.
2. Click the General tab.
3. Right click mouse button to bring up the popup menu.
4. Choose "Customize ToolBox".
5. Locate and check VitaminCtrl class.

6. Close the dialog.
7. The Vitamin control icon will appear in the General panel. You could use the control hereafter.

Create a Member Variable for VitaminCtrl

This is only applicable to VC++. Right click on the control and choose **ClassWizard**. In the ClassWizard window select the **Member Variables** tab and add a new member variable for the object, for example **m_VitCtrl**.

Set Visual Server contact information to Get Live

Video/Audio streams

The basic operation is to get live video and audio streams from VS server. When using the control, you must set the RemoteIPAddr,ServerModelType, UserName and Password properties. Optionally, you could also set the HTTP port. Note, you could assemble all the information into Url property, but we recommend you to use the former way because different model needs different URL path. With correct settings above, you could now call Connect method to start the download of Video and Audio streams.

VC++

Add the following sample code in your **OnInitDialog** function so that the program will start showing live images and playing waves as soon as it starts. The IP address in the RemoteIPAddr must be the correct IP address to a visual server.

```
m_VitCtrl.SetUserName("root");
m_VitCtrl.SetPassword("root");
m_VitCtrl.SetRemoteIPAddr("192.168.0.100");
m_VitCtrl.SetServerModelType(2);          // we assume the model is 6000 server
m_VitCtrl.Connect();
```

VB

Add the following sample code in your **Form1_Load** function so that the program will start showing live images and playing waves as soon as it starts. The IP address in the RemoteIPAddr must be the correct IP address to a visual server.

```
VitaminCtrl1.UserName = “root”
VitaminCtrl1.Password = “root”
VitaminCtrl1.RemoteIPAddr = “192.168.0.100”
VitaminCtrl1.ServerModelType = esrv456KServer          ‘ we assume the model is 6000 server
VitaminCtrl1.Connect
```

C#

Add the following sample code in your **Form1_Load** function so that the program will start showing live images and playing waves as soon as it starts. The IP address in the RemoteIPAddr must be the correct IP address to a visual server.

```
axVitaminCtrl1.UserName = “root”;
axVitaminCtrl1.Password = “root”;
axVitaminCtrl1.RemoteIPAddr = “192.168.0.100”;
        // we assume the model is 6000 server
axVitaminCtrl1.ServerModelType = VITAMINDECODERLib.EserverModelType.esrv456Kserver;
axVitaminCtrl1.Connect();
```

Control PTZ camera attached to server

If you use camera that supports PTZ function on server or connect to a PTZ enabled IP camera, you can control the camera through SendCameraCommand method. Note that you can change the URL for PTZ control by setting PtzURL property, but usually there is no need to do this. And the control also has SendCameraCommandMap method that can be used to move the camera to the coordinate you assign (**Note: some external camera might not support this**). As for the Pan speed and Tilt speed, you could use SendCameraControlSpeed method to achieve this. They all use the same URL shown below.

VC++

```
// There is a default value, usually, you don’t have to set it
m_VitCtrl.SetPtzUrl("/cgi-bin/ camctrl.cgi ");
m_VitCtrl.SnedCameraCommand("up");    // move the camera up
```

VB

```
‘ There is a default value, usually, you don’t have to set it
VitaminCtrl1.PtzUrl = "/cgi-bin/ camctrl.cgi "
VitaminCtrl1.SnedCameraCommand "up"          ‘ move the camera up
```

C#

```
// There is a default value, usually, you don’t have to set it
axVitaminCtrl1.PtzUrl = "/cgi-bin/ camctrl.cgi ";
axVitaminCtrl1.SnedCameraCommand("up");     // move the camera up
```

Save current image to a file

You can save the current image to a local file using Bitmap or Jpeg format (the latter format is only available if the connection receives jpeg from server). Please refer to SaveSnapshot for more details. The following code saves an image to D:\image.jpg file in BMP format. When calling this method, the connection must have been established.

VC++

```
m_VitCtrl.SaveSnapshot(2, “D:\image.jpg”);
```

VB

```
VitaminCtrl1. SaveSnapshot  ePicFmtBmp , “D:\image.jpg”
```

C#

```
axVitaminCtrl1. SaveSnapshot (
     VITAMINDECODERLib.EPictureFormat.ePicFmtBmp, “D:\image.jpg”,);
```

Get current image data in your program

You can also handle the image data in the program by calling GetSnapshot.
The following code illustrates how to get a bitmap from the control. Note that you have to connect to Visual Server before calling GetSnapshot.

VC++

```
VARIANT vData, vInfo;
m_VitCtrl.GetSnapshot(2, &vData, &vInfo);
…     // use the buffer

VariantClear(&vData);          // must release the buffer
VariantClear(&vInfo);          // must release the buffer
```

VB

```
Dim vData As Variant
VitaminCtrl1.GetSnapshot   ePicFmtBmp, vData
… ‘ use the buffer
```

C#

```
object objData;
byte [] byData;

axVitaminCtrl1.GetSnapshot(VITAMINDECODERLib.EPictureFormat.ePicFmtBmp, ref vData);
byaData = (byte []) vData;
…     // use the buffer
```

How to load Vitamin control dynamically

Sometimes, it is more flexible to create a control dynamically rather than create the control statically. This section depicts how to create control dynamically.

VB

In VB the loading is by calling Form1.Controls.Add. The second argument for the call

is the new name for the newly created control. You have to use this name as the identifier when delete the control later on. You have to add the Vitamin control into your project first. **Note that, you have to uncheck the “Remove information about unused ActiveX Controls” setting in Project Properties\Make tab to avoid VB from removing the information of the control at run time.**

Load the control

```
Dim WithEvents X As VitaminCtrl

Set X = Form1.Controls.Add("VITAMINDECODER.VitaminCtrl", "Test", Form1)

X.Visible = True
X.Move 1, 100 * 15, 300 * 15, 200 * 15
X.Url = "http://root:123@192.168.1.90/cgi-bin/video.vam"
X.ServerModelType = esrv3KServer
X.Connect
```

Unload the control

```
Form1.Controls.Remove "Test"
```

C#

In C# the loading is by calling Controls.Add. The argument is a newly created control by calling “new”. You have to save the control handle to be used for remove later on. You have to add Vitamin control into you project first. And add a dummy Vitamin control to your window form to let the IDE add references needed for the control (The dummy control could be removed right after added).

Load the control

```
System.Resources.ResourceManager resources =
        new System.Resources.ResourceManager(typeof(Form1));
this.axVitaminCtrl1 = new AxVADECODERLib.AxVaCtrl();

this.axVitaminCtrl1.Enabled = true;
this.axVitaminCtrl1.Location = new System.Drawing.Point(184, 72);
this.axVitaminCtrl1.Name = "axVitaminCtrl1";
```

```
this.axVitaminCtrl1.OcxState =
  ((System.Windows.Forms.AxHost.State)(resources.GetObject("axVitaminCtrl1.OcxState")));
this.axVitaminCtrl1.Size = new System.Drawing.Size(192, 192);
this.axVitaminCtrl1.TabIndex = 1;
Controls.Add(axVitaminCtrl1);

axVitaminCtrl1.Visible = true;
axVitaminCtrl1.Left = 1;
axVitaminCtrl1.Url = "http://root:123@192.168.1.90/cgi-bin/video.vam";
axVitaminCtrl1.Connect();
```

Unload the control

```
Controls.Remove(axVitaminCtrl1);
axVitaminCtrl1= null;
```

C++

In C++ the loading is by calling new operator for the control class and then Create function for the object just created. You have to save the control memory to be used for remove later on. You have to add Vitamin control into you project first. And add a dummy Vitamin control to your dialog or window and add the event-handling template if you need to handle the event. To handle batch events for multiple controls, you need to add ON_EVENT_RANGE macro manually. In the sample code, you could see the syntax.

Load the control

```
m_lpVitamin = (CVitaminCtrl*)new CVitaminCtrl();

m_lpVitamin->Create("test",WS_CHILD|WS_VISIBLE,CRect(3,3,200,200),this,10001);

m_lpVitamin->SetAutoServerModelType(TRUE);

m_lpVitamin->SetUserName("root");
m_lpVitamin->SetPassword("123");
m_lpVitamin->SetRemoteIPAddr("192.168.1.240");
m_lpVitamin->SetCurrentControlCam(1);
m_lpVitamin->Connect();
```

Unload the control

```
m_lpVitamin->CloseConnect();
delete m_lpVitamin;
```

2.2 Application Sample Code

In the shipment, there are several sample codes. The following list the function for the sample code.

Vbtest

This project is the sample code written in VB 6.0. It demonstrates most of the function in this control. The functions include:

- Live streaming / Single jpeg
- Counting statistics data – Frame rate, Video bit rate, Audio bit rate, Reconnect count, connection time, total connection time
- Set DO
- Get DI by http command
- Motion Edit mode
- Misc. Options setting
- Recording
- AVI conversion
- InputPacket
- Server setting functions – Refresh, set, get, update
- Server Date/Time setting function
- Http command function
- Events - Video/Audio Decoded data notification, DI/DO notification, Packet notification
- Save snapshot in file

Vctest

This project is the sample code written in VC++ 6.0. It demonstrates most of the function in this control. The functions include:

- Live streaming / Single jpeg
- Counting statistics data – Frame rate, Video bit rate, Audio bit rate, Reconnect count, connection time, total connection time
- Set DO
- Get DI by http command

- Motion Edit mode
- Misc. Options setting
- Recording
- AVI conversion
- InputPacket
- Server setting functions – Refresh, set, get, update
- Server Date/Time setting function
- Http command function
- Events - Video/Audio Decoded data notification, DI/DO notification, Packet notification
- Save snapshot in file
- Get snapshot in memory and how to access the byte array
- How to assign AVI file by application

CSharptest

This project is the sample code written in C#. It demonstrates most of the function in this control. The functions include:

- Live streaming / Single jpeg
- Counting statistics data – Frame rate, Video bit rate, Audio bit rate, Reconnect count, connection time, total connection time
- Set DO
- Get DI by http command
- Motion Edit mode
- Misc. Options setting
- Recording
- AVI conversion
- InputPacket
- Server setting functions – Refresh, set, get, update
- Server Date/Time setting function
- Http command function
- Events - Video/Audio Decoded data notification, DI/DO notification, Packet notification
- Save snapshot in file

dynaloadcsharp

This project is the sample code written in C#. It demonstrates how to dynamically load the control into the program.

dynaloadvb

This project is the sample code written in VB 6.0. It demonstrates how to dynamically load the control into the program.

dynaloadvc

This project is the sample code written in VC++ 6.0. It demonstrates how to dynamically load the control into the program.

vcstring

This project is the sample code written in VC++ 6.0. It demonstrates how to get the string table for english.

testcsharpmdi

This project is the sample code written in C#. It demonstrates how to use control in MDI architecture.

Chapter 3

Application Program Interface Reference

This chapter contains the API function calls for the VA Control.

3.1 Enumeration

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

EAudioCodecType

List Member

Name	Value	Description
eAuCodecLow	1	Sample rate is 8K bps. Used in 6K servers.
eAuCodecMobile	2	Sample rate is also 8K. Used in RTSP servers.
eAuCodecStandard	3	Sample rate is 16K bps. Used in 3K server and 6K servers.
eAuCodecStereo	4	Sample rate is 16K/32/44.1K bps. Used in RTSP servers

Description

This enumeration is used when get current audio CODEC type.

EAVIRecordStatus

List Member

Name	Value	Description
eAVINone	0	The control is current connected or the control is not able to perform AVI conversion now.
eAVIStop	1	The AVI conversion is not running now.
eAVIRecord	2	The AVI conversion is currently running.

Description

This enumeration is used when get current AVI conversion status.

EControlStatus

List Member

Name	Value	Description
ctrlStopped	0	The control is now without any connection.
ctrlConnecting	1	The control is now connecting to remote server.
ctrlRunning	2	The streaming or single jpeg connection to server has been established.
ctrlDisconnecting	3	The connection is now disconnecting.

Description

This enumeration is used with the ControlStatus property.

EClickEventHandler

List Member

Name	Value	Description
clickNone	0	Doesn't not handle when user click on control.
clickHandleSelf	1	The control will move the camera to be centered on when user click. This value is working only for new firmware that support click on image.
clickSendEvent	2	The control will fire an event with coordinate to notify that user click on the control.
clickHandleSendEvent	3	This value is the combination of clickHandleSelf and clickSendEvent. This value is working only for new firmware that support click on image.

Description

This enumeration is used with the ClickEventHandler property.

EConnectionType

List Member

Name	Value	Description
eConnVideo	1	This is the video connection.
eConnAudio	2	This is the audio connection.

Description

This enumeration is used when connection status (OK/Broken) event fired.

EConnProtocol

List Member

Name	Value	Description
eProtNone	0	The control is currently not connecting to visual server.
eProtUDP	1	Use UDP as the connecting protocol.
eProtTCP	2	Use TCP as the connecting protocol.
eProtHTTP	3	Use HTTP as the connecting protocol. HTTP protocol does not support audio.
eProtMulticast	4	Use multicast to receive the streaming data. This is only valid for 7K servers and the firmware must support this.

Description

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

EControlButtonState

List Member

Name	Value	Description
ebutDigitalZoom	1	The digital zoom button.
ebutAVIConvert	1<<1	The AVI conversion button.
ebutTalk	1<<2	The talk button in two-way button.
ebutRtspPlayStop	1<<3	The play and stop buttons in RTSP model.
ebutPlayVolume	1<<4	The volume control buttons for audio enabled models (non-2K models)
ebutMicVolume	1<<5	The microphone control buttons for two-way model.
ebutRtspSlider	1<<6	The slider for RTSP model operates in file playback mode.

Description

This enumeration is used to specify the control bar outlook. It controls each button’s display or hide state. Note: server model type will also affect the button’s display state.

EControlType

List Member

Name	Value	Description
eCtrlNoCtrlBar	0	This setting asks the control not to display control bar.
eCtrlNormal	1	The control will show control bar and the image together
eCtrlMotion	2	The control will be shown in motion detection edit-mode. In such mode, no control bar will be shown.
eCtrlMaskEdit	4	The control will be shown in private mask edit-mode. In such mode, no control bar will be shown. This is only useful for VS2403.

Description

This enumeration is used to specify the control UI type. Note: if the control type is switched dynamically, the setting could not be updated if the connection is not re-established.

ECoordinateType

List Member

Name	Value	Description
eCoordAbsolute	1	The coordinate value is related to the upper-left corner of the map.
eCoordRelative	2	The coordinate value is related to current camera position.

Description

This enumeration is used when control the camera by map coordinates.

EDBRecordEventType

List Member

Name	Value	Description
eMDAlertWin1	1	Motion is detected for the first motion detection window.
eMDAlertWin2	2 (1<<1)	Motion is detected for the second motion detection window.
eMDAlertWin3	4 (1<<2)	Motion is detected for the first third detection window.
eDILow1	256 (1<<8)	Digital input 1 is low (the recording will keep until DI is not low)
eDILow2	512 (1<<9)	Digital input 2 is low (the recording will keep until DI 2 is not low). For 4 Channel model only.
eDILow3	1024 (1<<10)	Digital input 3 is low (the recording will keep until DI 3 is not low). For 4 Channel model only.
eDILow4	2048 (1<<11)	Digital input 4 is low (the recording will keep until DI 4 is not low). For 4 Channel model only.
eDIHigh1	65536 (1<<16)	Digital input 1 is high (the recording will keep until DI 1 is not high)
eDIHigh2	131072 (1<<17)	Digital input 2 is high (the recording will keep until DI 2 is not high). For 4 Channel model only.
eDIHigh3	262144 (1<<18)	Digital input 3 is high (the recording will keep until DI 3 is not high). For 4 Channel model only.
eDIHigh4	524288 (1<<19)	Digital input 4 is high (the recording will keep until DI 4 is not high). For 4 Channel model only.
eDIRise1	2097152 (1<<21)	Digital input 1 is changed from low to high. For 4 Channel model only.
eDIRise2	4194304 (1<<22)	Digital input 1 is changed from low to high. For 4 Channel model only.
eDIRise3	8388608	Digital input 1 is changed from low to high.

	(1<<23)	For 4 Channel model only.
eDIRise4	16777216 (1<<24)	Digital input 1 is changed from low to high. For 4 Channel model only.
eDIFall1	33554432 (1<<25)	Digital input1 is changed from high to low.
eDIFall2	67108864 (1<<26)	Digital input2 is changed from high to low. For 4 Channel model only.
eDIFall3	134217728 (1<<27)	Digital input3 is changed from high to low. For 4 Channel model only.
eDIFall4	268435456 (1<<28)	Digital input4 is changed from high to low. For 4 Channel model only.

Description

This enumeration is used in EventTypes property.

EDBStatusCode

List Member

Name	Value	Description
eStatusDiskFull	1	Disk is full when recording. The recording will be stopped when this status is called back.
eStatusDBRepairFinish	2	The database repair is finished. The lParam of the event means whether the repair success or not. Nonzero means success, 0 means not.
eStatusLocRepairFinish	3	The location repair is finished. The lParam of the event means whether the repair success or not. Nonzero means success, 0 means not.
eStatusNeedRepair	4	The location needs to be repaired. This often happens when recording procedure found that the location has consistency problem, such as some file are corrupted or deleted by other program.
eStatusRecordStart	5	Sent to users when the location starts to record. Often used when event recording to notify users that a new event happens and the recording starts.
eStatusRecordStop	6	Sent to users when recording stops.
eStatusRecordStop	7	The control is not set to cycle recording, and the size of location limitation has been reached.

Description

This enumeration is used in OnRecordStatus event.

EDisplayTimeFormat

List Member

Name	Value	Description
eTimeFmtNormal	0	The normal 24 hours format
eTimeFmtTwelves	1	12 or 24 hours format decided by the system setting in regional control panel. For 12 hours, the time marker is always “AM”/”PM” no matter what language the OS is.
eTimeFmtUser	2	12 or 24 hours format decided by the system setting in regional control panel. For 12 hours, the time marker is the same as the system setting.

Description

This enumeration is used in DisplayTimeFormat property.

EMediaType

List Member

Name	Value	Description
eMediaNone	0	There is no media now.
eMediaVideo	1	Only video data is meaningful for the control.
eMediaAudio	2	Only audio data is meaningful for the control.
eMediaAV	3	Both video and audio are meaningful for the cotnrol.

Description

This enumeration is used to specify the media to be got from servers.

EPanelBtnStyle

List Member

Name	Value	Description
eBtnAuto	0	The control will detect the server type automatically
eBtnPauseStop	1	The RTSP server is new model, both Pause and Stop button are shown
eBtnStopOnly	2	The server is old model. Only stop button is used.

Description

This enumeration is used only for RTSP servers. It determines the outlook of the control bar.

EPictureFormat

List Member

Name	Value	Description
ePicFmtJpeg	1	JPEG format.
ePicFmtBmp	2	Bitmap format. The first scanline is on bottom.
ePicFmtYUV	3	YUY2 format. It's ordering is Y1U1Y2V2Y3U3Y4U4...
EPicFmtRaw24	4	RGB24 format. The first scanline is on top.
EPicFmtIYUV	5	Planar format. All Ys follow by all Us and then all Vs
EPicFmtYV12	6	Planar format. All Ys follow by all Vs and then all Us

Description

This enumeration is used when get image from control or for video data notification. For snapshot, the jpeg mode might need to encode the decoded data to jpeg again. The performance might be bad.

EPTZEnableFlag

List Member

Name	Value	Description
eptzPTZOnOff	1	The camera in server has PTZ capability. For server that connects to fix camera, this bit is 0. And if this bit is 0, all other bits below are also 0.
eptzPTZBuiltIn	2	If the camera is built in or external. 0 for external, 1 (value 2) for built-in. For older firmware version. This bit is not set. Please update the firmware if necessarry
eptzPTZPan	4	If the camera supports Pan capability? 0 for no, 1 (value 4) yes. For older firmware version. This bit is not set. Please update the firmware if necessarry.
eptzPTZTilt	8	If the camera supports Tilt capability? 0 for no, 1 (value 8) yes. For older firmware version. This bit is not set. Please update the firmware if necessarry
eptzPTZZoom	16	If the camera supports Zoom capability? 0 for no, 1 (value 16) yes. For older firmware version. This bit is not set. Please update the firmware if necessarry
eptzPTZFocus	32	If the camera supports Focus capability? 0 for no, 1 (value 32) yes. For older firmware version. This bit is not set. Please update the firmware if necessarry

Description

This enumeration is used when retrieve the eCfgSystemPTZEnable1 setting. The setting is an integer that contains the combination of the above bits.

ERegistryRoot

List Member

Name	Value	Description
eRegLocalMachine	0	The registry key root is set to local machine.
eRegCurrentUser	1	The registry key root is set to current user.

Description

This enumeration is used in RegkeyRoot property.

EServerConfig

List Member

Name	Value	Description
eCfgSystemResetSystem	0	Should the server reboot after config.ini is updated by calling UpdateServerConfig.
eCfgSystemSerialNumber	2	The serial number for the server. This is the MAC address of the server. It's a read-only property.
eCfgSystemCurrentDate	3	The current date value for the server. Format is YYYY/MM/DD It's a read-only property.
eCfgSystemCurrentTime	4	The current time value for the server. Format is hh:mm:ss. It's a read-only property.
eCfgSystemFirmwareVersion	6	The firmware version string. It's a read-only property.
eCfgSystemPTZEnable1	8	The PTZ enable property. Users could cast it into a number. For older firmware, there is no such property in config.ini, so the value is always 0. Upgrade the firmware if necessary. It's a read-only property.
eCfgSystemUserName	14	Retrieve or set the user name for the 20 system allowed users. The lExtraIdx is used to identify the user to be set. 0 is always for the root user.
eCfgSystemUserPwssword	15	Retrieve or set the user password for the 20 system allowed users. The lExtraIdx is used to identify the user to be set. 0 is always for the root user.
eCfgVideoCaptionText	101	Retrieve or set the video caption text.

Description

This enumeration is used when retrieve or set the server configuration.

EServerModelType

List Member

Name	Value	Description
esrv2KServer	0	The server is a 2K with 1 channel model.
esrv3KServer	1	The server is a 3K model.
esrv456KServer	2	The server is a 4/5/6K model.
esrv7KServer	3	The server is a RTSP model.
esrv2KServer4Ch	4	The server is a 2K with 4 channels model.

Description

This enumeration is used to control the connectivity of the control to various servers. Users must specify correct server model type to get the control works correctly.

ESpeedType

List Member

Name	Value	Description
ePanSpeed	1	This is the pan speed for camera control
eTiltSpeed	2	This is the tilt speed for camera control.

Description

This enumeration is used when control the camera speed.

EStreamingOption

List Member

Name	Value	Description
eStOpStreaming	1	Use streaming to get the media.
eStOpSingleJpeg	2	Get single jpeg from server by CGI. It would be slower and all the extra information such as DI/DO and motion information that are carried in stream would not be available in such case.

Description

This enumeration is used to specify the streaming option when playing live media.

ETalkBtnStyle

List Member

Name	Value	Description
eTalkBtnToggle	1	The talk button in two-way connection is toggle type. It means that click on the talk button will enable talk session. The session will continue until users click the button again.
eTalkBtnPush	2	The talk button in two-way connection is push type. It means that users must keep on pressing the talk button to continue talk session. Once release, the talk session is ended.

Description

This enumeration is used to specify the talk button type. It’s only for two-way models.

EVideoCodecType

List Member

Name	Value	Description
eViCodecMJpeg	1	Video codec is motion jpeg
eViCodecSP	2	Video codec is simple profile mode
eViCodecSHM	3	Video codec is short header mode

Description

This enumeration is used when get current video CODEC type.

EVideoQuality2K

List Member

Name	Value	Description
evqua2KMedium	1	Video quality is medium
evqua2KStandard	2	Video quality is standard
evqua2KGood	3	Video quality is good
evqua2KDetailed	4	Video quality is detailed
evqua2KExcellent	5	Video quality is excellent

Description

This enumeration is used when adjust the video quality for current connection (Only for 2K servers). It takes effect by re-establish the connection. These definitions are the same as those defined in server’s home page.

EVideoSize2K

List Member

Name	Value	Description
evsz2KHalf	1	Video size is half (176x120 NTSC, 176x144 for PAL)
evsz2KNormal	2	Video size is normal (352x240 NTSC, 352x288 PAL)
evsz2KDouble	3	Vide size is double (704x480 NTSC, 704x576 PAL)

Description

This enumeration is used when adjust the video size for current connection (Only for 2K servers). It takes effect by re-establish the connection. These definitions are the same as those defined in server’s home page.

3.2 Properties

AudioBitRate

Retrieve the audio bit rate of current connection. This is only applicable to those models that support audio and the connection contains also audio data.

Type

Long

This is the audio bit rate in Bps (bytes per second).

Attribute

R/O

Remarks

To use this property, the application must set a timer to retrieve this value. Or the bit rate will be the average bit rate from connection starts. Once retrieved, the accumulated data bytes will be reset. And the next time when this property is called, the returned value is calculated from last retrieval time to current.

AutoAVISettings

When convert the live stream to AVI file, should the control use the default settings or use the settings users selected. The settings include: video and audio compressors, video frame rate, video width and height. For the automatic mode, the control will try to decide the proper value. The default value is True.

Type

Boolean

True means to use default settings, False means not.

Attribute

R/W

Remarks

The proper value is per control base, so if there are two or more controls convert AVI at the same time, the estimation might be incorrect.

AutoReconnect

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

Type

Boolean

True means to reconnect automatically, False means not.

Attribute

R/W

Remarks

No matter what value this property is set, the control will send out a connection-broken event. The event receiver should not try to do reconnection if this value is set to TRUE since the control itself will handle the reconnection.

AutoServerModelType

Decide whether the control should recognize the server model type automatically. With this setting, the users could now specify only the IP/User name/Password to get connected to server (though if the server is 4 channels model, the channel index must be set separately)

Type

Boolean

True means to get server model type automatically, False means not.

Attribute

R/W

Remarks

To decide the server type, the control needs to talk with server before connecting to for streaming. So it will take more time to get connected. After the server type is known, a OnServerModeType event will be fired (no matter what current server model is). Application could use this to change the UI setting.

AVIFilePathName

This property is used for application to specify the name of next generated AVI file. The name should include full path name. This property is only used when AVIManualNaming is set to true. When AVIManualNaming is set to true, the control will fire the event OnRequestAVIFileName when a new AVI file name is needed.

Type

String

This is the file name set by application.

Attribute

R/W

Remarks

This name could point to any directory that control could access. The application should ensure that the control have right to write to the target directory, because the control dose little error handling for the correctness of the file path name.

AVIManualNaming

This property tells the control if the AVI file name is given by user or generated by control itself. The default false as previous version, the control use the PC time and prefix to generate file name whenever it needs file name. And the file will be put under AVIPath. But if this property is set to true, the control will fire event OnRequestAVIFileName when a new AVI file name is needed. Application could give the file name by setting the property AVIFilePathName.

Type

Boolean

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

Attribute

R/W

Remarks

If application set this property to true and forgot to implement the event, the control will fall back to generate file name automatically after about 80 milliseconds of waiting. This is to prevent the whole control blocks forever.

AVIMaxFileSize

This property is used to set the maximum AVI file size. If both this property and AVIMaxFileTimeLength are set, the first limitation that reaches would be used.

Type

Long

The minimum value allowed is 500K bytes. The maximum possible value is 2G. Set the property to 0 could record the file without auto changing file. (But AVI has maximum size limitation for about 4G.)

Attribute

R/W

Remarks

Note that the size is not exactly. So the final size could somehow larger than the set size. But it will not be larger too much. So if application needs exact size constraint, please set the size with margin. For example, if the size should not be larger than 100M, set the property to 99M could ensure that the size never over 100M.

AVIMaxFileTimeLength

This property is used to set the maximum time in seconds an AVI file would contains. If both this property and AVIMaxFileSize are set, the first limitation that reaches would be used.

Type

Long

The minimum allowed value is 5 seconds. The maximum is 2G seconds though you seldom will use such large value. Set this value to 0 could turn off time limitation.

Attribute

R/W

Remarks

Note that the time period in file would be smaller or equals to the set seconds. It is in server time scale not in PC time scale. So it might not as what you expect if the network condition is bad.

AVIPath

The property contains the default path for the AVI files. Each time when users start AVI recording, a new file named by using the recording date/time will be created under this directory.

Type

String

This is the default recording path.

Attribute

R/W

Remarks

If this path contains multiple layers of directories, the control will ensure the path to be created properly. If the directory could not be created, the recording won't start.

AVIStatus

Retrieve the AVI conversion status for the control. User could use this property

Type

EAVIRecordStatus

The enumeration for the AVI status

Attribute

R/O

Remarks

AVIVideoFrameRate

This property is to specify or retrieve the video frame rate used to convert A/V data in database to AVI file.

Type

Long

The permitted range for this value is from 1 to 30.

Attribute

R/W

Remarks

This frame rate could be different from the actual frame rate recorded in the database. The control will insert dummy frames or drop frames to meet this frame rate for the final AVI files. This means the video data in AVI files will not be played in fast or slow motion effect no matter what value this property is set to.

Requirements

This property is used only when AutoAVISettings is set to False.

AVIVideoHeight

This property is to specify or retrieve the video height for the final AVI file. This value could be different from the video height of the actual video frame in database.

Type

Long

The value should be a multiple of 8. If not, some video compressor would have problem. The maximum allowed value is 2048, but in some video compressor, the maximum is smaller than 2048, in such case, the conversion will be failed. Please change to a smaller value if this happens.

Attribute

R/W

Remarks

This property is only valid if the AVIVideoSizeByStream is set to False. If AVIVideoSizeByStream is set to be True, it's ignored. All the video frame would be stretched to meat this height if the size is different.

Requirements

This property is used only when AutoAVISettings is set to False.

AVIVideoSizeByStream

This property is to determine if the video height and width is set according to the first video frame size when conversion or not.

Type

Boolean

True means to use the first video frame size as the output video size in the AVI file. All subsequent video frames would be stretched to the same size as the first frame is. False means to use the width and height specified in properties AVIVideoHeight and AVIVideoWidth.

Attribute

R/W

Remarks

When this property is set to True, AVIVideoHeight and AVIVideoWidth are ignored.

Requirements

This property is used only when AutoAVISettings is set to False.

AVIVideoWidth

This property is to specify or retrieve the video width for the final AVI file. This value could be different from the video width of the actual video frame in database.

Type

Long

The value should be a multiple of 8. If not, some video compressor would have problem.

Attribute

R/W

Remarks

This property is only valid if the AVIVideoSizeByStream is set to False. If AVIVideoSizeByStream is set to be True, it's ignored. All the video frame would be stretched to meat this width if the size is different.

Requirements

This property is used only when AutoAVISettings is set to False.

BitmapFile

This property let users set the bitmap to be shown when the control is not connected to any server device.

Type

String

The name of the bitmap file to be loaded

Attribute

R/W

Remarks

If the server is connecting or connected to server, the bitmap file will not be shown. The file name could start with “http://”. In such case, the control will retrieve the file from remote server and display it after finishing.

CircularMode

This property decides if the control should delete the oldest file in current location if the file system full or the location size limitation reaches.

Type

Boolean

True means to be circular, False means not.

Attribute

R/W

Remarks

This property should be used very careful, or it could cause lost of data.

ClickEventHandler

Tell the control how to response to the mouse click event.

Type

EClickEventHandler

HandleSelf is the default setting.

Attribute

R/W

Remarks

Please refer to EClickEventHandler for the possible value of this property.

ControlID

This is the control ID used to write or read registry.

Type

Long

This is the control ID

Attribute

R/W

Remarks

If the control ID is 0, the registry is read from or written to the default registry root. If the value is not 0, the registry is read from or written to a sub-key under the registry root. The sub-key name is constructed base on the control ID.

ControlPort

Tell the control the control port to use to communicate with the Visual Server.

Type

Long

This is the port number.

Attribute

R/W

Remarks

For 2K servers, this property is not meaningless. For 3/4/5/6K servers the control port is retrieved automatically. For 7K servers, this value is important if the SDP port of server is not default value (554).

ConnectionProtocol

Set the connection protocol used when connecting to server.

Type

EConnProtocol

The available protocol is listed in EConnProtocol.

Attribute

W/O

Remarks

Once set, the control will restart the connection if it is currently connecting to server. But there is no guarantee that the connection will use the protocol assigned, it could switch to other protocol if necessary. User could get the CurrentProtocol to know the protocol actually used.

ConnectionTimeout

Set or get the timeout value used when connect to server. The default value is 25 seconds.

Type

Long

The value is in milliseconds.

Attribute

R/W

Remarks

ControlButtonOpts

Get or set the display state of the buttons on control bar. In other word, you could show or hide each button set individually.

Type

Long

This is the bit-wise Or-ed value of the type defined in EControlButtonState

Attribute

R/W

Remarks

Besides this property, the server model type also affects the control bar buttons' state. The following list the default button state for each model type:

- 2K (both 1 channel and 4 channel): Digital zoom (show), AVI (hide)
- 3K: Digital zoom (show), AVI (hide), Play volume (show)
- 4/5/6 K: Digital zoom (show), Talk (show), AVI (hide), Play volume (show), Mic Volume (show)
- 7K: Digital zoom (show), AVI (hide), Play/Pause/Stop (show), Play volume (show), Slider (show if file base playback)

ControlStatus

Retrieve the control status. The control will maintain its status during operation.

Type

EControlStatus

This is the status for the control.

Attribute

R/O

ControlType

This property is used to change the different UI supported by the control. This property is better not to be changed after control created. That is, if the control is used in IE, the value should be set in parameter list, and if used in non-script language, the property should be set in designed time.

Type

EControlType

This is the type for the control. If it is changed at runtime, the control will switch to the new interface.

Attribute

R/W

Remarks

For client setting interface, the ServerModelType property does not affect the outlook of the control.

CurrentAudioCodecType

Read from local the codec type uses by the server.

Type

EAudioCodecType

The element will be updated after media received.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server and the connection contains audio media.

CurrentAudioPort

Read from local the audio port uses by server to provide audio stream service.

Type

Long

This is the port number.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server and the connection contains audio media.

CurrentAudioProtocol

Read from local the audio protocol uses to connect to server.

Type

EConnProtocol

The enumeration now contains four elements: None, HTTP, UDP, and TCP.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server and the connection contains audio media.

CurrentControlCam

Read or set the current control camera index. This property is meaningful for 4 -channel model.

Type

Long
the camera index.

Attribute

R/W

Remarks

For 1 channel model, this value is always treat as 1.

CurrentMediaType

Read from local the current media type that is available for the connection to server.

Type

EMediaType

This is media type. The value is one of the value in EMediaType.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server.

CurrentProtocol

Read from local the protocol uses to connect to server.

Type

EConnProtocol

The enumeration now contains four elements: None, HTTP, UDP, and TCP.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server.

CurrentVideoCodecType

Read from local the video codec type uses by the server.

Type

EVideoCodecType

The elements are listed in EVideoCodecType.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server and the connection contains video media.

CurrentVideoPort

Read from local the video port uses by server to provide video stream service.

Type

Long

This is the port number.

Attribute

R/O

Remarks

This value is meaningful only after connecting to the server and the connection contains video media.

DatabasePath

Get the database path that is currently set to the control.

Type

String

This is a string that represents the database path. The path format is the same as what you use under windows explorer.

Attribute

R/O

DBHierarchy

Get or set the way the media files are saved under location path. It determines if sub-directory is created to hold the media files. This property is process-wide. It means if you have more than one controls in your system, change the setting in one control will affect all other controls.

Type

Boolean

True means to create sub-directory, this is the default value. False means not.

Attribute

R/W

Remarks

In FAT32 partition, there is a file number limitation problem. Set this property to true will solve this problem because now each directory will not hold too many files. Sub-Directory solution also solve the problem that if there are large amount of files in the location, use explorer to browse that directory will spend “LONG” time (maybe half an hour).

Deblocking

Get or set the de-blocking mode when decode video stream.

Type

Boolean

True means to de-block the video image, False means not. Default value is False.

Attribute

R/W

Remarks

De-blocking could solve the blocky problem for video stream in low bandwidth, but it would consume more CPU power when decode. Please set this property carefully because it might cause the system to be insensitive for UI request because of CPU busy on decoding.

DecodeAV

Get or 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. False means not decode.

Attribute

R/W

Remarks

The default value is to decode audio and video data. If the value is changed to false, no audio would be heard and no video data would be shown on screen. But the screen would still show the server time and location name if the Display flag is not turned off.

DigitalInURL

Set or get the URL for digital input.

Type

String

This URL could include or exclude the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

DigitalOutURL

Set or get the URL for digital input/output.

Type

String

This URL could include or exclude the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

DigitalZoomEnabled

Set or get the digital zoom enabled property. This property controls if the digital zoom function is enabled when displaying video.

Type

Boolean

True means the digital zoom function is enabled. False means it’s disabled.

Attribute

R/W

Remarks

It is possible to enable/disable digital zoom function even if the edit interface is not shown.

DigitalZoomEnableChk

Set or get the property that controls if the “enable digital zoom” check box is shown or hidden in the zoom control panel.

Type

Boolean

True means check box is shown. False means it’s hidden.

Attribute

R/W

Remarks

When the check box is hidden, the digital zoom enabling or disabling could only be set through the property DigitalZoomEnabled. Users will not able to disable or enable digital zoom on UI.

DigitalZoomFactor

Set or get the digital zoom factor property. This property controls the proportion of the video frame to be zoomed.

Type

Long

The range is between 100 and 400. Value outside this range would be normalized to 100 when setting.

Attribute

R/W

Remarks

The size of the viewable window is (Real Video Size * 100 / this factor). And the position of the viewable window might change when set a new factor. The rule is to enlarge or shrink the window centered at the original rectangle center. If any side of the viewable window is outside the real video, the window is moved to align the real video frame at that side.

DigitalZoomX

Set or get the x coordinate of the left-upper corner of the viewable window. This and the DigitalZoomY property decide which portion of the video is viewable on screen.

Type

Long

The value range is between 0 and the width of half mode video size. So it depends on which kind of lens the server uses.

Attribute

R/W

Remarks

The control will adjust the value if the value is out of range.

DigitalZoomY

Set or get the y coordinate of the left-upper corner of the viewable window. This and the DigitalZoomX property decide which portion of the video is viewable on screen.

Type

Long

The value range is between 0 and the height of half mode video size. So it depends on which kind of lens the server uses.

Attribute

R/W

Remarks

The control will adjust the value if the value is out of range.

Display

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

Type

Boolean

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

Attribute

R/W

Remarks

For computer that has slower graphic card, this switch would be very helpful to prevent the system loading from being filled up.

DisplayMotionFrame

Set or get the switch to turn on or turn off the display of motion detection triggered frame (red rectangle).

Type

Boolean

True means to show the motion rectangle when triggered, False means to hide the rectangle. The default value is True.

Attribute

R/W

Remarks

The default value is True.

DisplayPeriod

Set or get the video display period. The setting value is the frame number that would be used to count before one frame is shown. For example, if the set value is 2, then the control will show one frame per two frames it receives. This property is useful to lower down the loading of the computer that running with several controls at the same time.

Type

Long

This is the frame numbers to be set. 0 or 1 means to show every frame, greater value means longer period before screen updated.

Attribute

R/W

Remarks

If the network speed is slow, for example if the control gets only one frame per second, this period count will be ignored. In other word, the DisplayPeriod will be disabled automatically in slow speed link.

DisplayTimeFormat

Set or get the format to display server time on control title.

Type

EDisplayTimeFormat

The enumeration elements are list in EDisplayTimeFormat.

Attribute

R/W

Remarks

For eTimeFmtTwelves format, the position of “AM”/”PM” is always after the hour/minute/second string. For the eTimeFmtUser, the position time marker is the same as what users see in Windows’ regional control panel.

DisplayErrorMsg

Set or get the switch to turn on or turn off the display of error message when error happens.

Type

Boolean

True means to show the message with message box. False means to disable display of error message. The default value is True.

Attribute

R/W

Remarks

Sometimes application need not to show users the error message could set this property to false. But now the message boxes are all timeout box that would be closed automatically after 20 seconds.

DDrawOnePass

Set or get the switch to decide the way to display video when in DirectDraw mode. The default value is False.

Type

Boolean

True means to show the video directly without shivering proof, False to provide shivering proof.

Attribute

R/W

Remarks

When drawing with DirectDraw, if the video is stretching, the video will be shivering when other window moving upon. The control will use some mechanism to cancel this shivering, but this mechanism won't work in some display card (the video looks bad if stretching or shrinking). The developing could turn this property one when the video quality is bad.

DrawHwnd

Set or get window handle that the video data will be drawn to. This is usefully if developers want to use this control in background and show the graph on existing application window.

Type

Long

This is the window handle of the target display window. This is actually a HWND type.

Attribute

R/W

Remarks

To restore the painting window to the original one, please set this property to 0 or call RestoreControlHandle.

EnglishString

Return the string by index in English. This is useful if you want to translate the string to other language. User could use the vcstring sample to retrieve the string table and translate them into other language. Use SetLangString, SetLangStringHex, SetGivenLangInfo to set the new language settings.

Type

String

The string value for specified entry. If the index is out of scope, the returned value is empty string. So if application is to retrieve the whole table, it could stop when get empty string.

Parameters

lIndex

[in] the index number for each entry in the string table. It starts from 0.

Attribute

R/O

Remarks

The string table could be changed as the control version grows. So the application should compare the string table when got newer version of control.

EventTypes

Set or get the types that will affect the event recording. The event type is combination of those bits defined in EDBRecordEventType.

Type

Long

The bits combination of those types defined in EDBRecordEventType.

Attribute

R/W

Remarks

For different servers, the number of DI information carried in video stream is also different. For DI bit 2, 3, 4, It’s only meaningful for 4-channel models.

ForceGDI

Should the control show video in GDI mode no matter the card supports DirectDraw or not.

Type

Boolean

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

Attribute

R/W

Remarks

GDI has a worse performance but better compatibility than DirectDraw upon display card. On the machine that has problem using DirectDraw, this property could be turned on.

ForceNonYUV

Should the control show video in DirectDraw mode (if possible) without YUV surface?

Type

Boolean

Set this value to True to force to display video in non-YUV mode. Default value is False.

Attribute

R/W

Remarks

In some card, using YUV to show video would lead to green screen. In such case, set this flag could still use the DirectDraw capability but solve the green screen problem.

FrameRate

Retrieve the frame rate of current connection. This is only applicable to those connection contains video data.

Type

Long

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

Attribute

R/O

Remarks

To use this property, the application must set a timer to retrieve this value. Or the frame rate will be the average frame rate from connection starts. Once retrieved, the accumulated frame number will be reset. And the next time when this property is called, the returned value is calculated from last retrieval time to current.

GDIUseStretchBlt

Set or get the property that controls the underlying display function when showing video. When this property is turned on, the video quality will be better, but the performance is worse.

Type

Boolean

True to gain better video quality, False to gain better performance.

Attribute

R/W

Remarks

This property is only usable when the ForceGDI is set to True or when the machine does not support DirectDraw. If the control runs in DirectDraw mode, this property is ignored. **Note: when the control size is small so that video is shrinking, the motion detection window would sometimes not seen when this property is not set to True. This is Windows API limitation. Turn on this property would somehow resolve this problem.**

GetMaskEditParmUrl

Get or set the mask edit parameters retrieving URL. Only works for VS2403

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

GetMDParmUrl

Get or set the motion detection parameters retrieving URL.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

HttpPort

Tell the control the http port to use to communication with the Visual Server.

Type

Long

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

Attribute

R/W

Remarks

Note this value is used to append to the several URLs properties if they are lack of the IP part.

HWnd

Get or set the drawing target window handle.

Type

Long (HWND)

The window handle used to draw.

Attribute

R/W

Remarks

Note this value is originally set to the handle of the window of the control. Once the value is changed, the original window will be hidden. To restore the original window, please refer to RestoreControlHandle.

IgnoreBorder

Get or set if the control should display the border when showing video. The border is the 5-pixel wide gray line 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

R/W

Remarks

When in Motion editing mode, the border could not be ignored

IgnoreCaption

Get or set if the control should display the caption when showing video. The border is the 20 pixels high text line above the video (and also above the border if any). When the value is True, the caption will be ignored. The default value is False.

Type

Boolean

Show or hide the caption

Attribute

R/W

Remarks

When in Motion editing mode, the caption could not be ignored.

IndexSize

Get or set the index number use in a media file. The media file size is controlled by two factors: MaxFileSize and IndexSize. Whenever one condition reaches, the media file will be closed, and a new media file will be opened and used.

Type

Long

The index size value.

Attribute

R/W

Remarks

IsRecording

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

Type

Boolean

The control is recording or not.

Attribute

R/O

Remarks

JpegQuality

Get or set the quality value for the jpeg notified by the control.

Type

Long

The value should be within 1-125. The larger value means worse quality.

Attribute

R/W

Remarks

If the control uses single jpeg mode as its streaming mode, this property would not affect the quality of the jpeg notified. Because the quality is determined when server generates the file. There is no reason to re-generate a jpeg with worse quality in client side.

JpegSecsPerFrame

Get or set the period to retrieve jpeg from server if the streaming mode is single jpeg.

Type

Long

This is the value in seconds. The default value is 1 second.

Attribute

R/W

Remarks

Please do not set the value too large, or the control would seem to be frozen.

JpegURL

Get or set the URL for single jpeg mode to retrieve file.

Type

String

This is the URL to retrieve jpeg. It could be full path or partial path. For partial path, the RemoteIPAddr must also be set.

Attribute

R/W

Remarks

The default value is good enough for use with the video server and IP camera. But if users intend to use with proxy server that could cache jpeg frames, this property could not changed to satisfy such condition.

Language

This property meaning is changed since 2.1.0.0. Now when application wants the control to refresh all the string with newly set value. He should set this value to ask the control to do the refreshment. The value is not important.

Type

String

Any value is OK.

Attribute

R/W

Remarks

If the language is not given, English will be used.

LeftTitleSpace

Get or set the space between left border and the “text on video” text when drawing caption of the control. The unit is in pixels.

Type

Long

The default value is 2.

Attribute

R/W

Remarks

Location

Get current location setting for database. A location is a subfolder that saves the media files for a certain server (We don’t recommend you to save several servers’ data under a same location).

Type

String

Any characters that are permitted for directory path are legal for location.

Attribute

R/O

Remarks

The length is dependent on database path length. Usually, you won’t give a location longer than 32 characters.

MaxFileSize

Get or set the maximum file size of one media file.

Type

Long

Attribute

R/O

Remarks

The unit is in bytes. So you could give at most 2G bytes for one file. But this is not recommended since once file system error, the whole file may be lost. But for size too small, there will be another problem that the maximum file numbers under one subdirectory may be reached easily.

MaxLocationSize

Get or set the maximum size a location allows to save media data when recording in circular mode. This property is ignored when circular mode is disabled.

Type

Long

Attribute

R/O

Remarks

The unit is in kilobytes. So you could give at most 1T bytes for one location. This would be sufficient for current file system. The default value is 30M bytes, and the minimum value is 5M bytes.

MDEditMode

Get or set the motion detection edit mode.

Type

Boolean

True to enable the edit mode, False to disable the edit mode. The former will show the window and meter, buttons.

Attribute

R/W

Remarks

When this property is changed to be true, the control will connect to the server to get the last setting of motion detection automatically. When the edit mode is turned off, all unsaved state will be lost.

MediaType

Set the media type for connection.

Type

EMediaType

This is the new media type used by this control.

Attribute

W/O

Remarks

Once change, the control will reconnect to the server dependent on the new value. For 2K server, the media type could be only video only. For 3K servers, the protocol must be matched for the media type. Http contains video only, TCP and UDP supports both audio and video. Other models support video only, audio only or A/V mode.

MediaRecord

Get the recording state of the control.

Type

Boolean

True means the recording flag of the control is turned on now. False means the control's recording flag is turned off.

Attribute

R/O

Remarks

To set the recording flag, please use StartMediaRecord and StopMediaRecord methods. This property is different from the IsRecording property. The latter one is true only when the control is writing data into hard disk. But his property is true after StartMediaRecord or StartMediaRecordEx is called.

MicMute

Get or set the microphone’s mute state when talk.

Type

Boolean

True means to mute the microphone. False means not.

Attribute

R/W

Remarks

Mute the microphone is truly to send server silent audio packet. This property will not affect other program to use the audio capture capability. And it also not affect other (ActiveX) instance’s mute state.

MicVolume

Get or set the microphone’s volume used when talk

Type

Long

This is the volume range from 0 to 100. 100 is the loudest level.

Attribute

R/W

Remarks

Because of limitation, this value is global to all control instances. And it also affect the Windows system’s audio capture volume level. In short, change the value will change the system’s setting.

NotifyImageFormat

Set or get the image type to be notified when new video data arrive. This property will be meaningful only if NotifyVideoData is set to be true.

Type

EPictureFormat

The available format is defined in EPictureFormat.

Attribute

R/W

Remarks

If the format is set the Jpeg, and the streaming option is set to normal streaming rather than single jpeg mode, the control needs to decode the video and re-encode it to jpeg. So the performance of the control would be even worse than just notify the decoded data (such as bmp24 or YUV). So please set this property carefully.

NotifyAudioPacket

Determine if the control should send control owner each audio packet received from network. The content of the notified data is a binary array that could be converted to the TmediaDataPacketInfo (defined in MainProfile SDK) structure (it's the pbyBuff member, parse the content could re-generate the structure).

Type

Boolean

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

Attribute

R/W

Remarks

The default value is False. This property is different from NotifyNewAudio. This one notify the un-decoded data, but the latter notify the PCM data format.

NotifyNewAudio

Determine if the control should send control owner new audio piece arrival event.

Type

Boolean

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

Attribute

R/W

Remarks

Because audio pieces are sent by byte array and it could be up to 6K bytes per piece. Notify too frequently would cause a performance penalty. The default value is False.

NotifyVideoData

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

Type

Boolean

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

Attribute

R/W

Remarks

Because video pieces are sent by byte array and it could be up to 1.2M bytes sometimes. Notify too frequently would cause a performance penalty. The default value is False.

NotifyVideoPacket

Determine if the control should send control owner each video packet received from network. The content of the notified data is a binary array that could be converted to the TmediaDataPacketInfo (defined in MainProfile SDK) structure (it's the pbyBuff member, parse the content could re-generate the structure).

Type

Boolean

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

Attribute

R/W

Remarks

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

PanelButtonStyle

Determine the way to display RTSP control buttons (play, pause and stop).

Type

EPanelBtnStyle

Specify the way to show the button.

Attribute

R/W

Remarks

For 3K servers that support RTSP protocol, the servers do not support pause function. In such case, the control button will show only the play and stop in the same button. For 7K servers, 2 buttons will be used to show the play/pause and stop function. If users know the server model, set this property to switch the button type. If users do not know, please use auto setting to let control detect the server type and change the button type.

Password

Get or set the password used for web page authentication.

Type

String

Attribute

W/O

Remarks

For security reason, this attribute is not retrievable.

PlayMute

Turn on or off the audio when playing.

Type

Boolean

True means to turn on the audio. And ‘false’ means to turn it off.

Attribute

R/W

Remarks

This property could be changed from control’s audio setting panel. But if the panel is dropped down and the value is changed by program, the value will not be reflected until the panel is closed and reopen again.

PlayVolume

Get or set the audio volume used when playing.

Type

Long

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

Attribute

R/W

Remarks

This property shares the same value set in the audio setting panel of the control bar. So if users change the value from UI, this property will be changed. As PlayMute is, change of the value will be reflected in next time the panel dropped down. Note: the volume value is for DirectSound play buffer, not global to the system. So it might happen that the sound is small even if the property is in maximum level. In such case, please adjust the global volume level from Windows' control panel.

PostEventTime

Get or set the time in seconds that would keep recording after a event happens.

Type

Long

This is the seconds to keep recording after event.

Attribute

R/W

Remarks

The default value is 5 seconds. The recording will not stop exactly on this value after event. If there is not other event happens before the recording stops, the recording will stop at N + M seconds. Where N is PostEventTime - 1 and M depends when the ‘I’ frame is seen. If the ‘I’ frame appears right after N, then the recording stops at N seconds. The maximum value for M is 4.

If there are other events happen before recording stop, the recording will not stop until the stop criteria match the time setting for the latest event.

PrebufferMemorySize

Get or set the pre-buffer size allocated for the control when event-recording mode is selected. From version 2.0.0.0, the control use memory to save the pre-buffer data. Properly set the value to meet the Pre-Event time. This property is process-wide. It means if you have more than one controls in your system, change the setting in one control will affect all other controls.

Type

long

this is the size in bytes. Minimum value is 256K and maximum value is 20M bytes

Attribute

R/W

Remarks

If the set value is too small, the pre-event data will not be kept, that means the recorded data will start from the first I after event happens. Set the value large would solve the problem but the memory usage for the system could increase dramatically.

PreEventTime

Get or set the time in seconds that the control keeps the data before event happens. This property might be affected by PrebufferMemorySize.

Type

Long

This is the seconds to keep recording before event.

Attribute

R/W

Remarks

The default value is 5 seconds. The control pre-buffers the data in temporary files. These files are removed if no event triggered after PreEventTime seconds. The kept media length will not be exactly PreEventTime seconds, but usually it will be longer. It’s possible to have shorter pre-event recording if the event happens right after the control starts to pre-buffer. Please do not set this value too large to avoid video blocking slightly when event record triggered.

PresetURL

Get or set the URL for save camera position.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

PtzURL

Get or set the URL for camera PTZ control.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

RecallURL

Get or set the URL for recall camera position.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

ReadSettingByParam

Get or set the property that directs the control to read parameter from registry of by parameter list if used in IE.

Type

Boolean

True means to accept setting by parameter list. ‘False’ means to use only registry setting.

Attribute

R/W

Remarks

This property is meaningful only if the control is used in IE. For other developing tool, the settings are always set by properties.

If MediaType property is updated, this flag will be automatically set to True.

ReadWriteTimeout

Get or set the timeout value for read/write from network.

Type

Long

This is the timeout value.

Attribute

R/W

Remarks

This timeout value is used after the connection is established to server. Please do not be confused with ConnectionTimeout.

ReconnectionWait

Get or set the time to wait before the control tries to reconnect to server.

Type

Long

Time to wait in milliseconds. Value less than or equals to 0 means to reconnect at once. Default value is 30000 milliseconds.

Attribute

R/W

RegkeyRoot

Get or set the registry root for saving control settings. It determines whether to save registry under HKEY_LOCAL_MACHINE or under HKEY_CURRENT_USER.

Type

ERegistryRoot

The default value is eRegLocalMachine, which means to save under HKEY_LOCAL_MACHINE.

Attribute

R/W

Remarks

This property lets users could save the setting under their own registry hierarchy.

RegSubKey

Get or set the registry base for saving control settings. The base starts from the root specify by RegkeyRoot. The default value is “Software\Vitamin\Fsdk”

Type

String

Attribute

R/W

Remarks

This property lets users could save the setting under their own registry hierarchy.

RemoteIDStr

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

Type

String

This is the string content.

Attribute

R/W

Remarks

This property is to increase the security when this control is used in IE that user could view the content of the program. In such program, because end users could look at the content of the code, the end user could then retrieve the password setting. With this property, the developer could first generate the encrypted value by using GenerateRemoteIDString.exe program contained in the installed package. The control will decrypted the user name and password at run time and hence keep user from retrieving the secret data.

Note if the set value of this property is not a correctly encrypted string, the user name and password property will remain unchanged.

RemoteIPAddr

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

Type

String

The format should be dotted IP, that is: ‘a.b.c.d’.

Attribute

R/W

Remarks

RemotePort

Get or set the remote control port of the Visual Server.

Type

Long

Attribute

R/W

Remarks

This property is set to hidden since it only means to be used in Internet Explorer.

RightTitleSpace

Get or set the space between right border and the “server time” text when drawing caption of the control. The unit is in pixels.

Type

Long

The default value is 2.

Attribute

R/W

Remarks

ServerConfig

Retrieve the server's configuration setting by index.

Type

String

This is the string value for an entry in the config.ini. Users could change the value to property according to the entry's property. For example, the ptzenable attribute is actually a number composed by bits.

Arguments

lCfgIndex

a Long type argument that specify the ID of the setting. The IDs are partly defined in the enumeration EServerConfig now. The full set of IDs could be found in ServerUtl module document (which is part of MainProfile).

lExtraInx

Some properties in the configuration file contain three layers. So extra index is needed to locate the final value. For example, the user name and password has three layers. Set this value to 0 to get the name of first user, 1 to get name of second user, …etc.

Attribute

R/O

Remarks

Before access this property, users must call RefreshServerConfig to retrieve server settings.

ServerConfigEntry

Retrieve the entry name of the server's configuration setting by index.

Type

String

This is the entry name. This value is unique for each index (except that some indexes are aliases for certain model)

Arguments

lCfgIndex

a Long type argument that specify the ID of the setting. The IDs are partly defined in the enumeration EServerConfig now. The full set of IDs could be found in ServerUtl module document (which is part of MainProfile).

Attribute

R/O

Remarks

Before access this property, users must call RefreshServerConfig to retrieve server settings.

ServerConfigSection

Retrieve the section name of the server’s configuration setting by index.

Type

String

This is the section name. This name is common to server index.

Arguments

lCfgIndex

a Long type argument that specify the ID of the setting. The IDs are partly defined in the enumeration EServerConfig now. The full set of IDs could be found in ServerUtl module document (which is part of MainProfile).

Attribute

R/O

Remarks

Before access this property, users must call RefreshServerConfig to retrieve server settings.

ServerModelType

Set the server model type of the target to be connected. This type must be correct so that the control could connect to the server successfully.

Type

EServerModelType

This is the he type of the server. When set, the control will also change the control bar outlook.

Attribute

R/W

Remarks

Please do not change the server type during connection. The ActiveX will response incorrectly if the server model changes during connection. Each model has different URLs for media streaming, single jpeg, PTZ URL, …ect. Please do not set the misc. URLs if connect to server or IP camera directly. The ActiveX will use default value for each model automatically. Those URLs could be changed if the connection from client to IP camera is through proxy or agent server.

For VS3101 user, the URL for streaming is different from the default value of 3000 servers. Please set the URL manually (Sample code contains such processing).

For PT users, the Preset location URL is different from other 3000 servers. If the call to add preset location is called before GetPtzPresetPosition, the functions (add or recall) will not work properly. If the target model is known when coding, the designers could set the PtzURL manually to solve such problem. Once the GetPtzPresetPosition is called, the add function could be working correctly.

SetMaskEditParmUrl

Get or set the mask edit parameters setting URL.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

SetMDParmUrl

Get or set the motion detection parameters setting URL.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

Stretch

Tell the control to stretch the graph to the size of the display window or not.

Type

Boolean

True means to stretch, False means not.

Attribute

R/W

Remarks

If the graph is not stretched, and the view part of the control is greater than the video frame, the frame would be put align to the upper left corner of the control. If the view part is smaller than the video frame, the center point of the view part and the frame would be put together.

StreamingOption

Set or get the streaming option for the control.

Type

EStreamingOption

The available options are listed in EStreamingOption.

Attribute

R/W

Remarks

When the streaming option is changed, the connection will be re-established.

TalkButtonStyle

Get or set the talk button style when the target server is two-way model.

Type

ETalkBtnStyle

This is the style of the button.

Attribute

R/W

Remarks

The default value is toggle.

TextOnVideo

Retrieve the text on video string that is shown on video. This is useful if users want to draw the graph themselves.

Type

String

The text on video received by control.

Attribute

R/O

Remarks

This value is only valid after connecting to server. And its value will be changed if someone changes the setting on server web page. Users could get this value periodically to check if it changes.

TitleBarColor

Get or set the background color used to draw the caption.

Type

OLE_COLOR

This is the color value. The default color is black (R:0, G: 0, B:0)

Attribute

R/W

Remarks

TitleTextColor

Get or set the text color used to draw the caption.

Type

OLE_COLOR

This is the color value. The default color is white (R:255, G: 255, B:255)

Attribute

R/W

Remarks

The color is only applicable to control in non-recording mode. When in recording mode, the text color is set to RED(R: 255, G: 0, B: 0). So please do not set TitleBarColor to RED or the caption would be not visible when recording.

UartURL

Set or get the URL for UART control.

Type

String

This URL could be included or excluded the host IP.

Attribute

R/W

Remarks

The default value works fine for Visual Server. Usually, you don’t need to change this value.

Url

Set or get the URL for video retrieving.

Type

String

This URL could include or exclude the host IP.

Attribute

R/W

Remarks

Each server type has its own default value. But the default value is used internal. If users set this property, the control will not use default value but the value set by users. Please set this value carefully.

UserDateFormat

Set or get the display server date format property.

Type

Boolean

True means to use the format set in Windows' regional control panel. False means to use the YYYY/MM/DD format.

Attribute

R/W

Remarks

The default value is False.

UserName

Set or get the user name for web authentication.

Type

String

Attribute

R/W

Remarks

This value could also be appended to the Url property. The control will parse and retrieve it from the string you given.

VideoBitRate

Retrieve the video bit rate of current connection. This is only applicable to those connection contains video data.

Type

Long

This is the video bit rate in Bps (Bytes per second).

Attribute

R/O

Remarks

To use this property, the application must set a timer to retrieve this value. Or the bit rate will be the average bit rate from connection starts. Once retrieved, the accumulated frame number will be reset. And the next time when this property is called, the returned value is calculated from last retrieval time to current.

VideoQuality2K

Set or get video quality for current connection to 2000 servers.

Type

EVideoQuality2K

Attribute

R/W

Remarks

When set, the connection will be re-established to reflect the change. The default value is good.

VideoSize2K

Set or get video size for current connection to 2000 servers.

Type

EVideoSize2K

Attribute

R/W

Remarks

When set, the connection will be re-established to reflect the change. The default value is normal size.

WheelEventHandler

Should the control starts mouse wheel event handler. The default setting is off. When mouse event handler is turned on, the control will send server a zoom in or zoom out command when mouse wheel is rolled. Roll upward will zoom in, roll downward will zoom out the view. The zoom in and zoom out command are only valid for cameras that have such capability. For those that do not, the command is sent but no effect.

Type

Boolean

True means to turn on mouse wheel event handle, and false means not.

Attribute

R/W

Remarks

When event handler starts. The control will grab the keyboard focus whenever the mouse pointer is moved inside the control. This is to ensure that the control could receive the wheel event. But it could not what application wants, turn off if this annoy you.

3.3 Methods

Because the OLE will raise exception when the return value of method is not S_OK, we always return S_OK for each method. And we add a parameter that is a pointer of long integer to hold the error code. Now when use these methods, just treat them as functions with return code of type long.

CloseConnect

Request the control to close the current connection. This command is similar to Disconnect except that the latter will wait until the connection is truly closed. But this command send the close connection command and return immediately. The control will keep performing the connection closing until it is done. So in short, this is a non-blocking mode disconnect command.

Syntax

```
HRESULT CloseConnect (                                    );
```

Return Value

Always S_OK.

Parameters

None

Remarks

This command is used to speed up application that needs to include more that one control in their user interface. When the application is closing, if it calls Disconnect one by one, the closing time would be a lot of time. With this method, the application could first call CloseConnect for each control, and then call the Disconnect one by one. Because CloseConnect is non-blocking, the connection disconnecting is performed in background. Hence the overall closing time would be almost equivalent to close a single control.

ChooseAVIAudioCompressor

This method is to choose the audio compressor used when converting AVI.

Syntax

HRESULT	**String** *bstrDialogTitle*	**);**
ChooseAVIAudioCompressor (	**Long** **plRet*	

Return Value

Always S_OK.

Parameters

bstrDialogFile

[in] This is the caption text for the audio compressor picker dialog.

plRe

[out] the return code of the function. 0 means success, others mean failed.

Remarks

After calling this method, the picked value will be applied to the control immediately (This does not apply to the conversion that starts before the settings change). And the settings are saved in the registry if the login user has right to access the registry. So next time when user opens this dialog again, the settings will keep the same no matter the control is restarted again or not.

Requirements

None.

ChooseAVIVideoCompressor

This method is to choose the video compressor used when converting AVI.

Syntax

```
HRESULT ChooseAVIVideoCompressor ( String bstrDialogTitle );
                                 ( Long *plRet
```

Return Value

Always S_OK.

Parameters

bstrDialogFile

[in] This is the caption text for the video compressor picker dialog.

plRe

[out] the return code of the function. 0 means success, others mean failed.

Remarks

After calling this method, the picked value will be applied to the control immediately (This does not apply to the conversion that starts before the settings change). And the settings are saved in the registry if the login user has right to access the registry. So next time when user opens this dialog again, the settings will keep the same no matter the control is restarted again or not.

Requirements

None

Connect

Connect to the Visual Server to start the download.

Syntax

HRESULT Connect (	**Long** **plRet*	**);**

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

If the connection is already established, this call will do nothing and return 0.

Requirements

The RemoteIPAddr, HttpPort, Url, ConnectionProtocol value should be ready before calling this method.

Disconnect

Disconnect to the Visual Server to stop the download. This call is blocking mode, so it will not returned until the connection is truly disconnected. To close the connection without being blocked, please use CloseConnect instead.

Syntax

```
HRESULT Disconnect (                                    );
```

Return Value

Always S_OK.

Parameters

None

Remarks

If the connection is not connected, this call will do nothing.

GetConnectionStatus

Get the connection status.

Syntax

```
HRESULT GetConnectionStatus ( Variant *pvData,
                              Long *plRet );
```

Return Value

Always S_OK.

Parameters

pvData

[out] The buffer that holds the returned data. It’s an array of type Long.

0: Reconnect Times

1: Live Time in seconds

2: Total Connection Time in seconds

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

GetDigitalIn

Get the digital input value for the specified port.

Syntax

HRESULT GetDigitalIn (	**Long *lPort*,** **Variant*** *pData*, **Long *** *plRet*	**);**

Return Value

Always S_OK.

Parameters

lPort

[in] Specify the port index of the digital input device to get data. Port index starts from 1. For 4-channel model, this is to indicate the DI index. For 1 channel model, this should be always 1.

plData

[out] The buffer that holds the returned data. 0 is for low-level signal. 1 is for high-level signal. Other values are reserved for future use. For script language to work properly, this parameter is set the Variant type.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This function works in blocking mode. It will not return until the data got or HTTP time out occurred.

GetPtzPresetPosition

Retrieve the list of name of preset positions of camera.

Syntax

```
HRESULT GetPtzPresetPosition ( VARIANT *pvData,
                               Long *plRet          );
```

Return Value

Always S_OK.

Parameters

pvData

[out] The buffer that holds the returned data. This value is actually an array of String that holds the preset position list.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

The returned pvData is an array of Bytes. You could get the size of the image by testing the size of the array.

GetSnapshot

Get the current decoded picture frame.

Syntax

HRESULT GetSnapshot (	**EPictureFormat** *eFormat*,	
	Variant * *pvData*,	
	Variant * *pvInfo,*	
	Long * *plRet*	**);**

Return Value

Always S_OK.

Parameters

eFormat

[in] The format of image caller needs.

pvData

[out] The buffer that holds the returned data. This value is actually an array of Byte that holds the image data.

pvInfo

[out] Lists the information for the image. It's an array of Variant.

0: Width (Long)

1: Height (Long)

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

If the given picture format is incorrect, error will be returned.

GetUartData

Get the data on the UART interface of server.

Syntax

HRESULT GetUartData (	**Variant** **pvData*,	
	Long *lReadLen*,	
	Long *lPort*,,	
	Long *lTimeout*,	
	Long * *plRet*	**);**

Return Value

Always S_OK.

Parameters

pvData

[in] The buffer that holds the returned data. This is a variant that contains a string that hold the hex string of the data read from COM port on Visual Server..

lReadLen

[in] This is the length user want. This value can't not be greater than 128.

lPort

[in, defaultval(1)] This value is the Uart port index for the reading action. For 4-channel model, it could be 1 or 2. For one channel model, only 1 is applicable. This value could be skipped under VB (but it only works for port 1 if not specified).

lTimeout

[in, defaultval(3000)] This value is the timeout value for the reading action. This value could be skipped under VB.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This function works in blocking mode. It will not return until the data got or HTTP timeout occurred. So the timeout value should not be set to high.

GetUartDataBinary

Get the data on the UART interface of server.

Syntax

```
HRESULT GetUartDataBinary (    Variant *pvData,
                               Long lReadLen,
                               Long lPort,,
                               Long lTimeout,
                               Long *plRet          );
```

Return Value

Always S_OK.

Parameters

pvData

[out] The buffer that holds the returned data. This is an array of Byte. VC users need to free the memory by calling VariantClear() function.

lReadLen

[in] This is the length user want. This value can't not be greater than 128.

lPort

[in, defaultval(1)] This value is the Uart port index for the reading action. For 4-channel model, it could be 1 or 2. For one channel model, only 1 is applicable. This value could be skipped under VB (but it only works for port 1 if not specified).

lTimeout

[in, defaultval(3000)] This value is the timeout value for the reading action. This value could be skipped under VB.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This function works in blocking mode. It will not return until the data got or HTTP timeout occurred. So the timeout value should not be set to high.

HttpCommand

Send a http command to server by either POST or GET.

Syntax

```
HRESULT HttpCommand (   String strUrlCommand,
                        Boolean bPost,
                        Boolean bReadData,
                        Variant *pvReadData,
                        Long *plRet             );
```

Return Value

Always S_OK.

Parameters

strUrlCommand

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

bPost

[in] Is the command to be sent as POST or GET. True to be sent by POST. False to be sent by GET.

bReadData

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

pvReadData

[out] This is a variant that contains a string to hold the retrieved data from server. The maximum data length this command could be handled is 50K. All data beyond this rage will be lost.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This function works in blocking mode. It will not return until the data got or

HTTP timeout occurred.

InputMediaPacket

Input the data packet received from network. The packet must be a packet output by remote Vitamin control's OnNewPacket or by DataBroker's A/V data callback.

Syntax

```
HRESULT InputMediaPacket(    Long lLength
                             Variant *pvPacket,
                             Long *plRet          );
```

Return Value

Always S_OK.

Parameters

lLength

[in] This is the packet length of the input data.

pvPacket

[in] This is a variant that contains the binary data part of a packet. Please refer remark section for more details.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

TMediaDataPacketInfo is a C structure. The pvPacket parameter of this method is actually the "pbyBuff" member of the structure. The lLength parameter describes the total length of the binary data. In other word, it's value equals to "dwBitstreamSize + dwOffset". The pvPacket could have larger size than lLength does. This could let application to reuse a buffer to pass data into this control.

InputMediaPacketX

Input the data packet received from network. The packet must be a packet output by remote Vitamin control's OnNewPacket or by DataBroker's A/V data callback. This method is designed for C/C++ application that could avoid COM wrapper when input packet.

Syntax

```
HRESULT InputMediaPacketX (   Long lLength
                              Long lDataPtr,
                              Long *plRet      );
```

Return Value

Always S_OK.

Parameters

lLength

[in] This is the packet length of the input data.

pvPacket

[in] This is an unsigned character pointer that holds the input data.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

TMediaDataPacketInfo is a C structure. The pvPacket parameter of this method is actually the "pbyBuff" member of the structure. The lLength parameter describes the total length of the binary data. In other word, it's value equals to "dwBitstreamSize + dwOffset". The pvPacket could have larger size than lLength does. This could let application to reuse a buffer to pass data into this control.

RecallPtzPosition

Move the camera to some preset position.

Syntax

HRESULT RecallPtzPosition (	**String** *strPosition*,	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

strPosition

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

plRet

[out] the return code of the function. 0 means success, others mean failed.

RefreshServerConfig

Force the control to retrieve the server setting (config.ini) from the control again. This operation must be called after the server's remote IP, user name and password are set.

Syntax

```
HRESULT RefreshServerConfig (    Long *plRet                    );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This operation is operated in blocking mode, so before the connection is finished or failed, the call will not return.

In some server model, the operation will be finished by FTP protocol. So it might not work in some environment that allows only HTTP protocol. In such case, please upgrade the firmware. Newer firmware always provides a CGI that could let users retrieve config.ini.

RepairDatabase

Repair the database when it could not opened correctly.

Syntax

HRESULT RepairDatabase (	**String** *strDatabase*,	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

strDatabase

[in] This is the full path name of the database path.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This function will create a new thread to repair the database, and once the repair finish, an event OnRecordStatus will be callback with eStatusDBRepairFinish status code, and a True or False for lParam. If the control is closing before the repair finish, the thread will be terminated and the database will be left in a inconsistent state (It's ok to call repair again to turn the database again in consistent state).

Application must be careful not to call repair database for the same database in more than one control at the same time. The result will be unpredicted.

If a database is open successfully, the call to this method will be failed with error code VS3ERR_DB_DATABASE_INITIALED. This is to prevent repairing a database that is under use.

When a control is repairing database, any call to database related functions will be failed with error code VS3ERR_DB_REPAIRING.

If a database could be opened and this method is called, an error code VS3ERR_DB_DONT_NEED_REPAIR is returned.

RepairLocation

Repair the location when it could not opened correctly.

Syntax

HRESULT RepairLocation (	**String** *strLocation*,	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

strLocation

[in] This is the location name.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This function will create a new thread to repair the location, and once the repair finish, an event OnRecordStatus will be callback with eStatusLocRepairFinish status code, and a True or False for lParam. If the control is closing before the repair finish, the thread will be terminated and the database will be left in a inconsistent state (It's ok to call repair again to turn the database again in consistent state).

Application must be careful not to call repair location for the same location in more than one control at the same time. The result will be unpredicted. Usually, if one control maps to one location, this won't be an issue.

An application should not call repair to various location at the same time. This will cause IO contention condition. The result is a very bad system performance.

Before calling this method, the database must be opened first. If a location is open successfully, the call to this method will be failed with error code VS3ERR_DB_LOCATION_OPENED. This is to prevent repairing a location that is under use.

When a control is repairing location, any call to database related functions will be failed with error code VS3ERR_DB_REPAIRING.

If a location could be opened and this method is called, an error code VS3ERR_DB_DONT_NEED_REPAIR is returned.

RestoreControlHandle

Restore the window handle for drawing to the control iteself.

Syntax

```
HRESULT RestoreControlHandle ( Long *plRet );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

SavePresetPosition

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

Syntax

```
HRESULT SavePresetPosition (    String strPosition,
                                Long *plRet             );
```

Return Value

Always S_OK.

Parameters

strPosition

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

plRet

[out] the return code of the function. 0 means success, others mean failed.

SaveSendMail

Save the current frame of image to a file specified. And then invoke the default mailer program to let users send out the image to remote users.

Syntax

```
HRESULT SaveSendMail (    EPictureFormat eFormat,
                          String strSendto,
                          Long *plRet                );
```

Return Value

Always S_OK.

Parameters

eFormat

[in] The format of image caller needs. The format of image caller needs.

strSendTo

[in] This is the ‘send to’ field in the mailer. It’s a default value. The value could be changed after the mailer program open.

plRet

[out] the return code of the function. 0 means success, others mean failed.

SaveSnapshot

Save the current frame of image to a file specified.

Syntax

HRESULT SaveSnapshot (	**EPictureFormat** *eFormat*,	
	String *strFileName,*	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

eFormat

[in] The format of image caller needs. The format of image caller needs.

strFileName

[in] This is the name of the file used to save the image.

plRet

[out] the return code of the function. 0 means success, others mean failed.

SendCameraCommand

Send server the camera control command.

Syntax

```
HRESULT SendCameraCommand ( String strCommand,
                            Long lTimeout
                            Long *plRet );
```

Return Value

Always S_OK.

Parameters

strCommand

[in] This is the command that supported by the server now:
right, left, up, down, home, tele, wide, near, far, auto, pan, patrol, stop, irisauto, open, close, custxxx

lTimeout

[in, defaultval(30000)] This is the timeout value for the first five command listed above. For VB users, it could also be skipped to use the default value.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

The command supported by this control may not work for some model. That's because the server does not support the method. For example, "pan" command will only works for PT model, but not for VS model.

SendCameraCommandMap

Send server the camera control command with relative or absolute coordinate.

Syntax

```
HRESULT SendCameraCommandMap ( Long lX,
                               Long lY,
                               Long *plRet );
```

Return Value

Always S_OK.

Parameters

lX

[in] The x coordinate related to the upper-left corner of the control or the x amount of movement related to current position.

lY

[in, optional] The y coordinate related to the upper-left corner of the control or the y amount of movement related to current position.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This method is only effective if the server is running with new firmware that supports click on image.

SendCameraControlSpeed

Send server the camera control command to control the camera movement speed.

Syntax

```
HRESULT SendCameraControlSpeed (  ESpeedType eType,
                                  Long lSpeed,
                                  Long *plRet          );
```

Return Value

Always S_OK.

Parameters

eType

[in] The speed type defined in ESpeedType.

lSpeed

[in] The speed value, the range should be 5~-5 include 0.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This method is only effective if the server is running with new firmware that supports pan/tilt speed setting.

SendDigitalOut

Send server the digital output to certain port.

Syntax

```
HRESULT SendDigitalOut (    Long lPort,
                            Long lValue,
                            Long *plRet     );
```

Return Value

Always S_OK.

Parameters

lPort

[in] Specify the DO index of the digital output device to set data. Index starts from 1. It is not used if the server contains only one port.

lValue

[in] 0 for low level signal, 1 for high level signal. Other values are reserved for future uses.

plRet

[out] the return code of the function. 0 means success, others mean failed.

SendUartCommand

Send server the command to certain COM port.

Syntax

```
HRESULT SendUartCommand (   String strCommand,
                            Boolean bFlush,
                            Long lPort,
                            Long *plRet         );
```

Return Value

Always S_OK.

Parameters

strCommand

[in] This is the command to send to UART on server. It's hex string converted from the binary command.

bFlush

[in, defaultval(0)] Should the server flush the COM port before sending data? The default value is False (default value is only working for VB)

lPort

[in, defaultval(1)] Specify the COM port index. For 4-channel model, it could be 1 or 2. For 1 channel model, it should be always 1. It could be skipped for VB users.

plRet

[out] the return code of the function. 0 means success, others mean failed.

SendUartCommandBinary

Send server the command to certain COM port.

Syntax

```
HRESULT SendUartCommandBinary (  Variant vCommand,
                                 Boolean bFlush,
                                 Long lPort,
                                 Long *plRet          );
```

Return Value

Always S_OK.

Parameters

vCommand

[in] This is the command to send to UART on server. It's an array of Byte.

bFlush

[in, defaultval(0)] Should the server flush the COM port before sending data? The default value is False (default value is only working for VB)

lPort

[in, defaultval(1)] Specify the COM port index. For 4-channel model, it could be 1 or 2. For 1 channel model, it should be always 1. It could be skipped for VB users.

plRet

[out] the return code of the function. 0 means success, others mean failed.

SetBitmapHandle

Set the bitmap to be shown when the control is not connected to any server.

Syntax

```
HRESULT SetBitmapHandle (   Long lBitmapHandle,
                            Long *plRet             );
```

Return Value

Always S_OK.

Parameters

lBitmapHandle

[in] This is the bitmap handle. Its value is a Win32 BITMAP handle. For example, in VB the Image property of Picture box contains a Handle sub property. And that's the BITMAP handle for the image.

plRet

[out] the return code of the function. 0 means success, others mean failed.

SetDatabasePath

Set the database path used by this control. This database is used only for recording.

Syntax

HRESULT SetDatabasePath (	**String** *strPath*,	
	Boolean *bAutoCreate,*	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

strPath

[in] This is the path points to when the database exists.

bAutoCreate

[in] Should the control create the database if it cannot find any media database at the path you specified?

plRet

[out] the return code of the function. 0 means success, others mean failed.

SetGivenLangInfo

Set the new language information. The control will use the new language for UI string.

Syntax

HRESULT SetGivenLangInfo (	**long** *lCodePage*,	
	Long *lFontSize,*	
	Long *lCharSet,*	
	Boolean *bPitch,*	
	Boolean *bSwiss,*	
	String *strFontName,*	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

lCodePage

[in] This is the code page for the new language. This value is important when translating the given string. In control, the strings will be translated between multi-byte and Unicode.

lFontSize

[in] the size of the font to be used to show the strings in control (non-message box part). Different language might need different font size. User could adjust this value to fit his need.

lCharSet

[in] The character set is useful when creating font. If the char set is not correct, the font might not be created successfully.

bPitch

[in] If the font should be created with pitch? Some language does need this.

bSwiss

[in] some language need this flag to be set.

strFontName

[in] The name of the font to be created.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

Note this function must be called before SetLangStringHex or SetLangString is called.

The following table is the table for known languages that is tested well for this control. If your language is not in the list, please try the parameters yourself.

Language	CodePage	FontSize	CharSet	bPitch	bSwiss	FontName
Traditional Chinese	950	14	0	False	False	"Verdana"
Simplified Chinese	936	14	134	False	False	“”
Japanese	932	16	128	True	True	“”
Germany	1250	14	0	False	False	"Verdana"
Czech	1252	16	238	False	False	“”
Spanish	1252	16	0	False	False	“”
French	1252	16	0	False	False	“”
Korean	949	14	129	True	True	“”

SetLangString

Set one string for new language by index. The string is in Unicode.

Syntax

HRESULT SetLangString (	**long** *lIndex*,	
	String *strValue,*	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

lIndex

[in] The index of the string to be set. If the index is out of range, VS3ERR_OUT_SCOPE would be returned.

strValue

[in] The string for the entry in Unicode.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

Note before calling this function, user must call SetGivenLangInfo because the code page must be known when set string. The control would do conversion between multi-byte and Unicode.

After all string entries are set, please set an arbitrary language value to Language property to let the control to refresh UI.

For the item that is not set, English will be used.

SetLangStringHex

Set one string for new language by index. The string is in multi-bytes and changed to hex value. The control will convert it back to multi-bytes and Unicode when needed.

Syntax

```
HRESULT SetLangStringHex (    long lIndex,
                              String strValue,
                              Long *plRet        );
```

Return Value

Always S_OK.

Parameters

lIndex

[in] The index of the string to be set. If the index is out of range, VS3ERR_OUT_SCOPE would be returned.

strValue

[in] The string for the entry in hex value.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

Note before calling this function, user must call SetGivenLangInfo because the code page must be known when set string. The control would do conversion between multi-byte and Unicode.

This function is provided to help the web base user that need to set language in html file.

After all string entries are set, please set an arbitrary language value to Language property to let the control to refresh UI.

For the item that is not set, English will be used.

SetLocation

Set the location for storing media data.

Syntax

HRESULT SetLocation (	**String** *strLocation*,	
	Boolean *bAutoCreate,*	
	Long **plRet*	**);**

Return Value

Always S_OK.

Parameters

strLocation

[in] This is the subdirectory under the database to store data.

bAutoCreate

[in] Should the control create the location if it can't find the matched location under current database?

plRet

[out] the return code of the function. 0 means success, others mean failed.

SetServerConfig

Update a setting to the local copy of server configuration file. This command will not update the setting on server. Users must call UpdateServerConfig to update all the changes to server.

Syntax

```
HRESULT SetServerConfig (   Long lCfgIndex,
                            Long lExtraIndex,
                            String strValue,
                            Long *plRet         );
```

Return Value

Always S_OK.

Parameters

lCfgIndex

[in] Specifies the ID of the setting. The IDs are partly defined in the enumeration now. The full set of IDs could be found in ServerUtl module document (which is part of MainProfile)

lExtraIdx

[in] Some properties in the configuration file contain three layers. So extra index is needed to locate the final value. For example, the user name and password has three layers. Set this value to 0 to get the name of first user, 1 to get name of second user, …etc

strValue

[in] This is the new value for the specified setting entry.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Requirements

The RefreshServerConfig must be called before this method could be called.

SetServerDateTime

Update the server's date and time. The date and time could not be set along, that is, the two strings must be both valid value.

Syntax

```
HRESULT SetServerConfig (   String strDate,
                            String strTime,
                            Long *plRet         );
```

Return Value

Always S_OK.

Parameters

strDate

[in] The new date value to be set. The format is YYYY/MM/DD. The month and day value must be always formatted as two-digit value.

strTime

[in] The new time value to be set. The format is hh:mm:ss. All the hour, minute and second must be formatted as two-digit value.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This method could be called before server is connected. But the remote server IP must be set before calling. Because it uses HTTP Post to update the server's date and time, if the call is made through Internet, there might be some time delay before the setting applied. So the result will be several second's lag for the time value.

Requirements

The RefreshServerConfig must be called before this method could be called.

StartAVIConversion

Start to record the AVI file. This method is only available when the control is connected.

Syntax

```
HRESULT StartAVIConversion (    Long *plRet                    );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

The AVI conversion is a time consuming operation because it needs to encode the raw data into specify codec. To speed update the conversion, users is recommended to install third party's codec such as MS[TR] Mpeg4 encoder.

StartMediaRecord

Set the MediaRecord flag to true. This function will check if all the setting is correct for recording, for example, the SetDatabasePath and SetLocation must be called before calling this method. This method is the same as calling StartMediaRecordEx with event record flag set to False.

Syntax

HRESULT StartMediaRecord (	**EMediaType** *eMediaType*, **Long** **plRet*	**);**

Return Value

Always S_OK.

Parameters

eMediaType

[in] Which media to record to the database? Currently, this parameter is not implemented and reserved for future.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

The recording will start after the control connects to server. So the return of this method doesn't always mean that the recording starts (It marks the recording flag internally). If you call this method during connecting mode, the recording starts immediately.

To let users knows that the control is recording. The caption of the control would turn RED when recording, and turn back to normal color after recording stops.

StartMediaRecordEx

Set the MediaRecord flag to true. This function will check if all the setting is correct for recording, for example, the SetDatabasePath and SetLocation must be called before calling this method. This method is to replace StartMediaRecord. Users could specify if the recording is normal continuous recording or event recording.

Syntax

```
HRESULT StartMediaRecordEx (    EMediaType eMediaType,
                                Boolean bEventRecord,   );
                                Long *plRet             );
```

Return Value

Always S_OK.

Parameters

eMediaType

[in] Which media to record to the database? Currently, this parameter is not implemented and reserved for future.

bEventRecord

[in] Whether the recording is normal continuous recording or a event driven recording. For more detail about event recording, please see the remarks section below.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

The recording will start after the control connects to server. So the return of this method doesn't always mean that the recording starts (It marks the recording flag internally). If you call this method during connecting mode, the recording starts immediately.

Event recording means the control will start recording when events are triggered. Users could select which kind of events to start recording by setting EventTypes. To prevent the recording contains no important information because the time when event triggered the video is not I frame, the control will pre-buffer a sequence of video frame. The length of time to pre-buffer could be set by PreEventTime. The recording will stop automatically after a period of time. The period could be adjusted by setting PostEventTime (For DI high or DI low, the recording continues until the DI state changes). If events are triggered continuously, the recording might last for a long time. This is to prevent losing important information because event might happen between the time recording stopped and the next I frame comes. In such case, that event would be lost (not recorded).

To let users knows that the control is recording. The caption of the control would turn RED when recording, and turn back to normal color after recording stops.

StartPacketInput

Start to input packet into the control. The input packet mode is designed to let users to receive data from remote by its own protocol, and thus could provide proxy like solution.

Syntax

HRESULT StartPacketInput (	**EmediaType** *eMediaType* **Long** **plRet*	**);**

Return Value

Always S_OK.

Parameters

eMediaType

[in] This is the media type of the data. The type is defined in EMediaType.

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

The input packet mode and the connection mode could not be coexistent. So if users call this method, any existing connection will be closed first.

For the control to play the video and audio correctly, the application must set correct media type. This is to let the control knows what media will be input in later method calls.

StopAVIConversion

Start the AVI conversion.

Syntax

```
HRESULT StopAVIConversion (    Long *plRet                );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

StopMediaRecord

Set the recording flag of the control to false.

Syntax

```
HRESULT StopMediaRecord (       Long *plRet                 );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

StopPacketInput

Stop the packet input mode. The control will clear the video to bitmap or show in blank.

Syntax

```
HRESULT StopPacketInput (       Long *plRet                 );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

UpdateServerConfig

Update the server configuration setting by upload the local copy. The local copy could be modified by calling SetServerConfig.

Syntax

```
HRESULT UpdateServerConfig (    Long *plRet                    );
```

Return Value

Always S_OK.

Parameters

plRet

[out] the return code of the function. 0 means success, others mean failed.

Remarks

This method always uses FTP to upload the server setting. Currently, if the running environment does not support FTP protocol (for example, the running PC is inside NAT, and FTP port is blocked), there is no way to work around, please open the FTP port blocking to make it work.

The control will default change the configuration file to prevent server to reboot after the configuration is updated. If users want to change some setting that would be effective only if server reboots, please update the reboot setting to true before calling this method.

Some setting could not be changed by this method because the server will not apply the setting read from configuration file, such as the server's date and time. That's server limitation. Please use HTTP command to update those kinds of setting.

Requirements

The RefreshServerConfig must be called before this method could be called.

3.4 Events

This control supports connection point. With this mechanism, control owner could receive certain events when certain condition happens. To receive these events, VC users should implement the event-sinking interface. Readers could find the example of how to implement the event-sinking interface using MFC in the sample codes. For those that don’t use MFC, please search on the Internet for the ATL implementation of sinking target. VB users could easily implement the events by click on **Procedures/Events Box** to insert the events.

OnAVIStatus

The control fires this event whenever the AVI conversion status is changed.

Syntax

HRESULT OnAVIStatus (	**EAVIRecordState** *eStatus*	**);**

Return Value

Please always return S_OK.

Parameters

eStatus

[in] the new status the control changes to.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnClick

The control fires this event whenever user clicks on the control by using mouse pointer and the ClickEventHandler is set to HandleSelf or HandleSelfSendEvent.

Syntax

HRESULT OnClick (	**long** *lX*,	
	long *lY*	**);**

Return Value

Please always return S_OK.

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.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnConnectionBroken

The control fires this event whenever the connection is broken by some reason other than user stop it.

Syntax

```
HRESULT OnConnectionBroken ( EConnectionType eConnType );
```

Return Value

Please always return S_OK.

Parameters

eConnType

[in] This parameter indicates the connection type of connection that is broken. Two values are available now: connAudio, connVideo.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnConnectionOK

Sent whenever the control connects to the Visual Server successfully.

Syntax

HRESULT OnConnectionOK (EConnectionType *eConnType,* **);**

Return Value

Please always return S_OK.

Parameters

eConnType

[in] This parameter indicates the connection type of connection that is broken. Two values are available now: connAudio, connVideo.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnDIDOAlert

The control fires this event whenever it gets the window alert event from video stream.

Syntax

HRESULT OnDIDOAlert (	**Long lChangeFlag,** **Long lDIDOValue**	**);**

Return Value

Please always return S_OK.

Parameters

lChangeFlag

[in] This parameter contains DI & DO change flag, the low word saves DI change flag, and the high word contains DO change flag. If bit 0 of low word is 1 means DI 1 changes, 0 means does not change. If bit 0 of high word is 1 mean DO 1 change (might be changed by other client), 0 means it does not change. The bit 1-3 of the low word has the similar meanings that apply to DI 2-4. This bit 1 of high word applies to DO 2. DI 2-4 and DO 2 are only available for 4-channel models.

lDODOValue

[in] This parameter contains DI & DO current value. The low word saves DI value, and the high word contains DO value. If bit 0 of low word is 1 means DI 1 is high, 0 means DI 1 is low. If bit 0 of high word is 1 mean DO 1 is high, 0 means DO 1 is low. The bit 1-3 of the low word has the similar meanings that apply to DI 2-4. This bit 1 of high word applies to DO 2. DI 2-4 and DO 2 are only available for 4-channel models.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnMDAlert

The control fires this event whenever it gets the window alert event from video stream.

Syntax

HRESULT OnMDAlert (	**Variant** **pvStatus*	**);**

Return Value

Please always return S_OK.

Parameters

pvStatus

[in, ref] This parameter contains a two-dimension array that holds the information for the three motion detection windows. vStatus[n][0] is a Boolean value that identifies if alert happens for the corresponding window. vStatus[n][1] is a Long value that hold the percentage of alert.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally. **Note: for the way to retrieve the array from the variant in various language, please refer to the sample code.**

OnNewAudioPiece

The control fires this event whenever it comes the new audio piece from the connection. Users must set the property NotifyNewAudio to True to receive this event.

Syntax

```
HRESULT OnNewAudioPiece (      Variant *pvPiece          );
```

Return Value

Please always return S_OK.

Parameters

pvPiece

[in, ref] This parameter contains the new arrival audio data. This parameter contains an array of two variants. The first one is a three-element long integer array to indicate time in second, millisecond, and the length of the audio data. The second one is the byte array contain the audio PCM data.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnNewPacket

The control fires this event whenever a new media packet is received from network. The packet could be video or audio. Users must set the property NotifyAudioPacket and NotifyVideoPacket to request audio or video packets.

Syntax

HRESULT OnNewPacket (	**Boolean** *bVideo* **Variant** **pvPiece*	**);**

Return Value

Please always return S_OK.

Parameters

bVideo

[in] Indicates if the packet is video or audio. True if it is video. False if it is audio.

pvPiece

[in, ref] This parameter contains the new arrival media data. This parameter contains an array of two variant. The first one is a three-element long integer array to indicate time stamp second, millisecond, and the size of the byte array. The second one is the byte array contain the video or audio un-decoded data.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnNewVideo

The control fires this event whenever a new image frame is received from server. If the NotifyVideoData property is not set to True, only bSignal, and the video time stamp is correct. This event is always fired even NotifyVideoData is not set to notify the new coming of video data.

Syntax

HRESULT OnNewVideo (	**Boolean** *bSignal,*	
	Boolean *bDecodedImg*	
	Variant **pvPiece*	**);**

Return Value

Please always return S_OK.

Parameters

bSignal

[in] Indicates if there are signal for the new image. It's only available for new version firmware

bDecodedImg

[in] Indicates if the notification contains data. Users should access the video data byte array only if this flag is set to True.

pvPiece

[in, ref] This parameter contains the new arrival video data. This parameter contains an array of two variant. The first one is a three-element long integer array to indicate time stamp second, millisecond, and the size of the byte array (only valid if bDecodeImg is True). The second one is the byte array contain the video decoded data (only valid if bDecodeImg is True).

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

Please do not access the size element and the byte array part of pvPiece if the bDecodedImg is False. This will produce unpredicted result.

OnRecordStatus

The control fires this event whenever error happens during recording.

Syntax

```
HRESULT OnRecordStatus ( Long lStatus
                         Long lParam    );
```

Return Value

Please always return S_OK.

Parameters

lStatus

[in] Indicates the status code for this notification. The status code is defined in EDBStatusCode. If the status is 1 means the disk is full, no space left for recording. Note, you have to handle the disk full event by stopping the recording, or you will get this event each time when network packets arrive.

lParam

[in] This argument is now not used. Reserved for future.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

OnRequestAVIFileName

The control would fire this event when it is going to generate a new AVI file. The application should response to this event by setting proper file name through the property AVIFilePathName. The application should set the name immediately after it got the event because the control is waiting for the new name there, if the application does not respond immediately, the whole control will be blocked.

Syntax

```
HRESULT OnRequestAVIFileName (      Variant *pvTime );
```

Return Value

Please always return S_OK.

Parameters

pvTime

[in, ref] This parameter contains the time value for the time that the file is going to be generated. Application could use this value to format the file name or it could use it's own naming rule to generate the new file name. The time is the packet time, not the PC time.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

Application must specify the file name through AVIFilePathName property. And application should not do too many works in the event, because the control is blocked before this event returned.

OnServerModelType

The control fires this event whenever the server model of the connected camera changes.

Syntax

```
HRESULT OnServerModelType (    EServerModelType Type        );
```

Return Value

Please always return S_OK.

Parameters

eType

[in] the new server model type in type EServerModelType.

Remarks

When implement this event function, the function should be declared as void. The return value here is used by OLE library internally.

3.5 Error Code List

The following is the error list for the control. The error code is returned by the last parameter of each method.

Code	Name	Meaning
1002	VS3ERR_HTTP_BLK_ERR	Error when doing http blocking operation, such as GetPtzPresetPosition, GetDigitalIn, GetUartData and HttpCommand.
1003	VS3ERR_CONNECT	Error when connecting to server. It is returned when calling Connect, RefreshServerConfig, SetServerDateTime, UpdateServerConfig.
1004	VS3ERR_ALREADY_START_INPUT	When call StartPacketInput and the control is already in input packet mode (has been called before but not stop yet).
1005	VS3ERR_INPUT_NOT_START	When call InputMediaPacket or InputMediaPacketX but the StartPacketInput is not called yet.
1101	VS3ERR_DB_PATH_INCORRECT	The database path is not correct. Either it doesn't exist or it points to a file rather than a path.
1102	VS3ERR_DB_NOT_EXIST	Database doesn't exist. When giving database path, caller could ask to create the database automatically or just open existing database. This error code is returned if callers want to open existing database, but it is not there.
1103	VS3ERR_DB_CR_FAILED	Failed to create database. Maybe the disk is full or because of permission denied.
1104	VS3ERR_DB_NOT_INIT	Database is not initialized.
1105	VS3ERR_DB_LOC_NOTFOUND	The specified location is not found. This is returned when callers don't want the control to create the location automatically and the location doesn't exist.
1106	VS3ERR_DB_LOC_CR_FAILED	Create location failed. The possible reasons are disk full or users have no permission to create directory or create file on the target

		directory.
1107	VS3ERR_DB_LOC_OTHER_ERR	Other location related error.
1110	VS3ERR_DB_LOC_NOT_INIT	The location is not initialized before calling StartMediaRecord or StartMediaRecordEx.
1117	VS3ERR_DB_DONT_NEED_REPAIR	Database does not need to be repaired. It's healthy.
1118	VS3ERR_DB_REPAIRING	Database is under repairing, no other database operation is permitted.
1119	VS3ERR_DB_LOCATION_OPENED	Location is opened. In such case, repairing database or repairing location are both not permitted.
1120	VS3ERR_DB_DATABASE_INITIALED	Database is opened. In such case, repairing database is not permitted.
1201-1215		Internal component Error
1301-1307		Memory related error
1401	VS3ERR_URL_MISS_DI	URL for retrieving DI status is empty.
1402	VS3ERR_DI_INCORRECT_FORM	The returned DI status page from server is not the format that we know. It's possible that the server is not in the list that this control supports.
1403	VS3ERR_PRESET_INCORRECT_FORM	The returned Preset location page from server is not the format that we know. It's possible that the server is not in the list that this control supports.
1404	VS3ERR_SNAPSHOT_FMT_UNSUPP	The snapshot output format is not in the supported list.
1405	VS3ERR_SNAPSHOT_FAILED	Snapshot failed.
1406	VS3ERR_URL_MISS_UART	URL for UART read/write is empty.
1407	VS3ERR_URL_MISS_PTZURL	URL for PTZ control is empty.
1408	VS3ERR_EXCEED_MAX_LEN	The given buffer length exceeds the maximum COM port read/write buffer length (128)
1409	VS3ERR_POINT_NOTIN_CTRL	The coordinate of the given point is not within control client area.
1410	VS3ERR_URL_MISS_DO	URL for DO setting is empty.
1411	VS3ERR_PARAM_INCORRECT	● The vData parameter for

		SendUartCommand or SendUartCommandBinary is not a byte array. ● At least one parameter for SetServerDateTime is empty string. ● InputMediaPacket detects that the given array is not a valid byte array.
1412	VS3ERR_URL_MISS_PRESET	URL for preset page is empty.
1413	VS3ERR_URL_MISS_VAM	URL for A/V stream is empty.
1415	VS3ERR_IPHOSTNAME_MISS	Does not give any IP or hostname info before calling Connect
1416	VS3ERR_SNAPSHOT_NOT_DECODE	Snapshot is requested but the DecodeAV property is set to False.
1417	VS3ERR_NOT_CONNECTED	Snapshot could not be got if not connects to server. This could also happen if SetServerConfig, SetServerDateTime and UpdateServerConfig are called before RefreshServerConfig is called.
1419	VS3ERR_INVALID_PACKET	The input packet method detects that the input packet format is not correct.
1420	VS3ERR_SAVE_REG_FAIL	Error happens when saving setting to registry. Maybe the user is not authorized to modify registry.
1421	VS3ERR_OUT_SCOPE	The language index is out of scope.
1422	VS3ERR_STRING_TOO_LONG	The string entry for certain index is too long. The maximum length is 255 character in multi-bytes.
1501-1503		File system error
3001		No match items