LiveView ActiveX API v1.3

Developer’s Manual

Target Audience

This document is for developers who would like to insert LiveView ActiveX component into third party applications.

This document is irrelevant for end users of surveillance systems.

Content

Chapter 1 What is LiveView ActiveX..................5

Chapter 2 Registration...6

Chapter 3 LiveView ActiveX Control Properties7

Variables..7
Connect...7
Disconnect..7

Chapter 4 LiveView ActiveX Control Methods..8

4.1 Show live video..8
Connect To Camera...8
Set Active View..8
Get Active View..8
Get Camera Index From View...8
Get Camera Count..9
Get Camera Name..9
Get Camera Model..9
Enable Audio...9
Play..9
Stop..10
Drop...10
4.2 Get raw image and snapshot...10
Get Raw Image Width...10
Get Raw Image Height..10
Get Raw Image..10
Snapshot...11
4.3 Perform PTZ control..11
Get Preset Count..11
Get Preset Name..11
PTZ Preset GO..11
Set Preset Number..12
PTZ Zoom Tele..12
PTZ Zoom Wide...12
PTZ Up..12

PTZ Down..13
PTZ Left..13
PTZ Right..13
PTZ Up Left...13
PTZ Up Right...13
PTZ Down Left...14
PTZ Down Right..14
PTZ Home...14
PTZ Patrol...14
PTZ Focus Near...14
PTZ Focus Far..15
4.4 Perform DI/DO control...15
Get DI Count..15
Get DO Count..15
Get DI Name..15
Get DO Name..15
Get DI Status...16
Get DO Status..16
Set DO Status..16
4.5 Display control and information feedback..................................16
Change Resolution..16
Set OSD Configuration..17
SetView1x1..18
SetView2x2..18
SetView3x3..18
SetView4x4..18
SetViewFullScreen..18
On Event..18

Chapter 1 What is LiveView ActiveX

LiveView ActiveX is used to perform liveview functions to a camera on a NVR/DVR based system. After connected, LiveView ActiveX can be utilized to perform numerous functions, which including:

1. Show live video
2. Get raw image and snapshot
3. Perform PTZ control
4. Perform DI/DO control
5. Display control and information feedback

LiveView ActiveX is designed to be a standalone component which can be easily inserted it into a third party application and further to connect to cameras. It provides necessary flexibility for wide range of usage yet with limited control code. It is more controllable in this way than creating a direct connection to the physical camera.

While using this component, the developers need not to know details about socket communication and image decoding. Instead, the developers simply control LiveView ActiveX component through its friendly interface.

First, LiveView ActiveX is capable to handles all communication with NVR/DVR based system, and will display related messages. Second, it provides complete camera control functions. Third, it can get raw image, which is beneficial for further processing. Fourth, DI/DO control can be performed. Finally, it provides display control and event control functions.

Architecture: Third Party SW LiveView ActiveX

Chapter 2 Registration

Before coding, you have to register LiveView ActiveX first. Double click “Register.bat” file under ActiveX folder. This action will register both “LiveActiveX.ocx” and “RPBActiveX.ocx”.

Chapter 3 LiveView ActiveX Control Properties

Variables

Variables listed below have to be declared before the connection is established.

BSTR IPAddress;

IP address of the live streaming server

BSTR UserName;

User name to login the live streaming server

BSTR Password;

Password for the user name

long Port;

TCP port of the live streaming server

Connect

BOOL Connect();

Connect to a specific server installed with NVR/DVR system.

Return Values:

TRUE return “TRUE” if connected to server successfully

FALSE return “FALSE” if failed to connect to server

Disconnect

BOOL Disconnect();

Disconnect from specific server installed with NVR/DVR system.

Return Values:

TRUE return “TRUE” if disconnected from server successfully

FALSE return “FALSE” if failed to disconnect from server

Chapter 4 LiveView ActiveX Control Methods

This chapter contains all LiveView ActiveX Control Methods. Methods are listed by functions and events are described towards the end of this section.

4.1 Show live video

Connect To Camera

BOOL ConnectToCamera(long iView, long iCameraIndex);

Connect to specified camera

Parameters:

iView which view to display (we can have 16 view in maximum, sequencing from 0~15)

iCameraIndex which camera to connect

Return Values:

TRUE return “TRUE” if the camera is connected successfully

FALSE return “FALSE” if failed to connect to the camera

Set Active View

BOOL SetActiveView(long iView);

Focus on specified view.

Notice: ConnectToCamera() has to be called first.

Parameters:

iView specify which camera is focused

Return Values:

TRUE return “TRUE” if camera is already connected to this view.

FALSE return “FALSE” if camera is not connected to this view.

Get Active View

long GetActiveView();

Get the focused view,

Return Values:

Active view index

Get Camera Index From View

long GetCameraIndexFromView(long iView);

Get camera's index from view

Parameters:

iView Which view is selected.

Return Values:

return "Camera index" of the specific view

return -1 if no camera connected

Get Camera Count

long GetCameraCount();

Get the aggregate number of all cameras in use

Return Values:

the aggregate number of all cameras

Get Camera Name

BSTR GetCameraName(long iCameraIndex);

Get the name of a specific camera

Parameters:

iCameraIndex Camera's Index

Return Values:

the name of a specific camera

Get Camera Model

BSTR GetCameraModel(long iCameraIndex);

Get the camera model

Parameters:

iCameraIndex Camera's Index

Return Values:

Camera model

Enable Audio

void EnableAudio(long iView, BOOL bEnable);

Enable the audio function of the camera

Parameters:

iView Which view is selected

bEnalbe return "TRUE" if audio function is enabled

return "FALSE" if audio function is disabled

Play

void Play();

Play a live video of an on active view from NVR/DVR based system

Stop

void Stop();

Stop a live video of an on active view from NVR/DVR based system

Drop

void Drop();

Drop a live video of an on active view from NVR/DVR based system

4.2 Get raw image and snapshot

Get Raw Image Width

long GetRawImageWidth(long iView);

Get the width of a specific camera's raw image

Parameters:

iView Get which view's raw image

Return Values:

the width of the raw image

Get Raw Image Height

long GetRawImageHeight(long iView);

Get the height of a specific camera's raw image

Parameters:

iView Get which view's raw image

Return Values:

the height of the raw image

Get Raw Image

VARIANT GetRawImage(long iView, long iVideoFormat);

Get raw image of a specific camera

Parameters:

iView Get which camera's raw image

iVideoFormat 0: YUV12

1: RGB

Return Values:

VARIANT.parray, which contains the whole raw image

Snapshot

BOOL Snapshot(long iView, LPCTSTR FileName);

Save the live video snapshot

Notice: SetActive() has to be called first

Parameter:

iView: Which Camera's snapshot

FileType: .bmp

.BMP

.jpg

.JPG

Return Values:

return "TRUE" if a snapshot is saved successfully

return "FALSE" if sailed to save a snapshot

4.3 Perform PTZ control

Get Preset Count

long GetPresetCount(long iView);

Get the count of the preset point(s)

Parameters:

iView which camera's total preset

Return Values:

the count of the preset point(s)

Get Preset Name

BSTR GetPresetName(long iView, long iPresetNum);

Get the name of a specific preset point

Parameters:

iView View index

iPresetNum Preset's number

Return Values:

the name of a specific preset point

PTZ Preset GO

void PTZPresetGO(long iView, long iPresetNum);

Go to a specific preset point

Parameters:

iView View index

iPresetNum Preset's number

Set Preset Number

BOOL SetPreset(long iView, long iPresetNum, LPCSTR PresetName);

Set a camera preset point of a specific view

Parameters:

iView View index

iPresetNum Preset number(-1 means delete all preset)

PresetName Specified preset name

Return Values:

return "TRUE" if preset point is set successfully

return "FALSE" if failed to set preset point

PTZ Zoom Tele

void PTZZoomTele(BOOL bStart);

Zoom in the camera

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to zoom in

1: start to zoom in

PTZ Zoom Wide

void PTZZoomWide(BOOL bStart);

Zoom out the camera

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to zoom out

1: start to zoom out

PTZ Up

void PTZUp(BOOL bStart);

Move the camera upward

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera upward

1: start o move the camera upward

PTZ Down

void PTZDown(BOOL bStart);

Move the camera downward

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera downward

1: start to move the camera downward

PTZ Left

void PTZLeft(BOOL bStart);

Move the camera to the left side

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera to the left side

1: start to move the camera to the left side

PTZ Right

void PTZRight(BOOL bStart);

Move the camera to the right side

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera to the right side

1: start move the camera to the right side

PTZ Up Left

void PTZUpLeft(BOOL bStart);

Move the camera to the up left side

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera to the up left side

1: start to move the camera to the up left side

PTZ Up Right

void PTZUpRight(BOOL bStart);

Move the camera to the up right side

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera to the up right side

1: start move the camera to the up right side

PTZ Down Left

void PTZDownLeft(BOOL bStart);

Move the camera to the down left side

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera to the down left side

1: start to move the camera to the down left side

PTZ Down Right

void PTZDownRight(BOOL bStart);

Move the camera to the down right side

Notice: SetActive() has to be called first

Parameters:

bStart 0: stop to move the camera to the down right side

1: start to move the camera to the down right side

PTZ Home

void PTZHome();

Move the camera to home position

Notice: SetActive() has to be called first

PTZ Patrol

void PTZPatrol(BOOL bStart);

Set camera to patrol through default preset point(s)

Notice: SetActive() has to be called first

Parameters:

bStart 0: start patrolling

1: stop patrolling

PTZ Focus Near

void PTZFocusNear(BOOL bStart);

Set the camera to focus on the nearer distance

Notice: SetActive() has to be called first

Parameters:

bStart 0: start to focus near

1: stop to focus near

PTZ Focus Far

void PTZFocusFar(BOOL bStart);

Set the camera to focus on the farther distance

Notice: SetActive() has to be called first

Parameters:

bStart 0: start to focus far

1: stop to focus far

4.4 Perform DI/DO control

Get DI Count

long GetDICount();

Get the count number of digital input

Return Values:

the count number of digital input

Get DO Count

long GetDOCount();

Get the count number of digital output

Return Values:

the count number of digital output

Get DI Name

BSTR GetDIName(long iInputIndex);

Get the name of a specific digital input

Parameters:

iInputIndex which input is selected.

Return Value:

the name of a specific digital input

Get DO Name

BSTR GetDOName(long iOutputIndex);

Get the name of a specific digital output

Parameters:

iOuputIndex which output is selected.

Return Value:

the name of a specific digital output

Get DI Status

Long GetDIStatus(long iInputIndex);

Check the status of a specific digital input

Parameters:

iInputIndex which input is selected.

Return Value:

return 0 if the status is low

return 1 if the status is high

Get DO Status

Long GetDOStatus(long iOuputIndex);

Check the status of a specific digital output

Parameters:

iOuputIndex which output is selected.

Return Value:

return 0 if the status is low

return 1 if the status is high

Set DO Status

void SetDOStatus(int iOutputIndex, long iValue);

Set the status of a specific digital output

Parameters:

iOutputIndex Which output needs to set enabled.

iValue 0: Low

1: High

4.5 Display control and information feedback

Change Resolution

void ChangeResolution(long iWidth, long iHeight);

Change the video window size.

Parameters:

iWidth video window width

iHeight video window height

Set OSD Configuration

void SetOSDConfig(long iCmd, LPCTSTR fontStyle, int iFontSize, int iColorR, int iColorG, int iColorB, BOOL bHasEdge, BOOL bBold, int iBColorR, int iBColorG, int iBColorB, iBKTransparency);

Set Camera's OSD Config.

Parameters:

iCmd 0x00: Disabled

0x01: Set Date

0x02: Set Time

0x04: Set Camera Number

0x08: Set Camera Name

0x10: Set Bitrate.

fontStyle OSD font style

iFontSize OSD font size

iColorR OSD font color R (0-255)

iColorG OSD font color G (0-255)

iColorB OSD font color B (0-255)

bHasEdge 0: Has edge

1: Doesn't has edge

bBold 0: Normal

1: Bold

iBColorR OSD background color R (0-255)

iBColorG OSD background color G (0-255)

iBColorB OSD background color B (0-255)

iBkTransparency OSD background transparency (0-255)

SetView1x1

void SetView1x1();

Set the view to a single window 0