Playback ActiveX API

Developer's Manual

Target Audience

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

Content

Chapter 1 What is Playback ActiveX

Playback ActiveX is used to perform playback functions on a NVR/DVR based system. After connected, Playback ActiveX can be used to perform a number of functions, which includes :

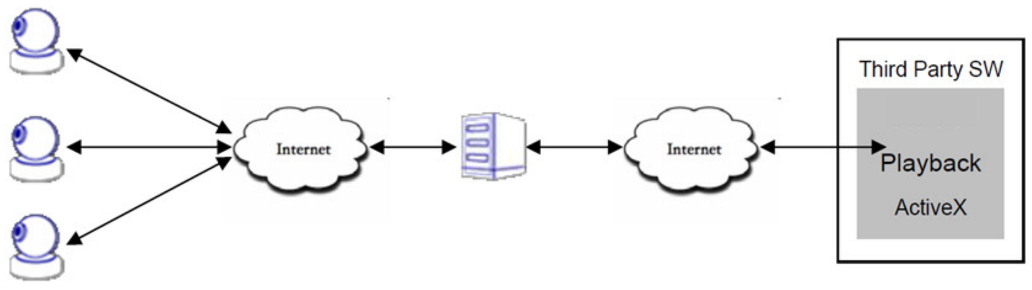
1. Get recording video
2. Perform Play Control
3. Export Format Settings
4. Display control and information feedback

Playback 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 Playback ActiveX component through its friendly interface.

First, Playback ActiveX is capable to handles all communication with NVR/DVR based system, and will display related messages. Second, it can get recorded video/image files and export tem the specific format. Finally, it provides display control and event control functions.

Architecture:



Chapter 2 Registration

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

Chapter 3 Playback ActiveX Control

Properties

Variables

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

BSTR IPAddress;

IP address of the remote playback server

BSTR UserName;

User name to login the remote playback server

BSTR Password;

Password for the user name

long Port;

TCP port of the remote playback 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 Playback ActiveX Control Methods

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

4.1 Get recording video

Open Record By Date Time

BOOL OpenRecordByDateTime(LPCTSTR StartDateTime, LPCTSTR EndDateTime, long iCameraLow, long iCameraHigh)

Open video record by the selected time and selected camera number and display on specific view

Parameters:

StartDateTime the start Date/Time point of the selected record video (YYYYMMDDHHMMSS)

EndDateTime the end Date/Time point of the selected record video (YYYYMMDDHHMMSS)

iCameraLow first 32 channel camera.

Ex: 0: no camera is selected

1: camera 1 is selected.

3: camera 1 and 2 are selected.

iCameraHigh last 32 channel camera.

Ex: 0: no camera is selected

1: camera 33 is selected.

3: camera 33 and 34 are selected.

Return Values:

TRUE :	return "TRUE" if opened recorded video successfully
FALSE:	return "FALSE" if failed to open recorded video

Set Active View

BOOL SetActiveView(long iView);

Focus on specified view.

Notice: OpenRecordByDateTime() 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:

the index of the active view

Get Camera Index From View

void GetCameraIndexFromView(long iView);

Get camera’s index of a specific view

Notice: The sequencing order is from left to right, top to bottom.

0	1
2	3

Parameters:

iView specify 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 count number of all cameras in use

Return Values:

the count 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 model of a specific camera

Parameters:

iCameraIndex Camera's Index

Return Values:

the model of a specific camera

4.2 Save video/image

Play

void Play();

Play a recorded video from a NVR/DVR based system

Pause

void Pause();

Pause a recorded video from a NVR/DVR based system

Stop

void Stop();

Stop from a NVR/DVR based system

Step Backward

void StepBackward();

Pause the recorded video and move the video backward one frame at a time

Step Forward

void StepForward ();

Pause the recorded video and move the time forward

Seek

BOOL Seek(LPCTSTR DateTime)

Move to specific Date Time point

Parameter:

DateTime Date time of selected record (YYYYMMDDHHMMSS)

Return Values:

TRUE : return "TRUE" if the time interval is valid.

FALSE: return "FALSE" if the time interval is nod valid.

Speed Up

void SpeedUp();

Accelerate the video's playing speed from 2 to 16 times faster

Speed Down

void SpeedDown();

Slow down the video's playing speed from 1/2 to 1/16 times slower

4.3 Play control

Save Avi

BOOL SaveAvi(long iView, LPCTSTR StartDateTime, LPCTSTR EndDateTime,
BOOL bExportAudio, BOOL bExportOSD, LPCTSTR FileName);

Save the selected time interval as an AVI format file

Notice: OpenRecordByDateTime() has to be called first

Parameters:

iView the selected camera index

StartDateTime the begin of the selected record video (YYYYMMDDHHMMSS)

EndDateTime the begin of the selected record video (YYYYMMDDHHMMSS)

bExportAudio 0: doesn't export audio

1: export audio

bExportOSD 0: doesn't export OSD

1: export OSD

FileName Saved file name

Return Values:

TRUE : return "TRUE" if the record exists

FALSE: return "TRUE" if the record doesn't exist

Save Asf

BOOL SaveASF(long iView, LPCTSTR StartDateTime, LPCTSTR EndDateTime,
BOOL bExportAudio, BOOL bExportOSD, long iProfile LPCTSTR FileName);

Save the selected time interval as an ASF format file

Notice: OpenRecordByDateTime() has to be called first

Parameters:

iView	the selected camera index
StartDateTime	the begin of the selected record video (YYYYMMDDHHMMSS)
EndDateTime	the begin of the selected record video (YYYYMMDDHHMMSS)
bExportAudio	0: doesn't export audio
	1:export audio

bExportOSD	0:doesn't export OSD
	1:export OSD
iProfile	0:Windows Media Video 8 for Local Area Network (256kbps)
	1: Windows Media Video 8 for Local Area Network (384kbps)
	2: Windows Media Video 8 for Local Area Network (768kbps)
	3: Windows Media 8 Fair Quality based VBR for Broadband
	4: Windows Media 8 High Quality based VBR for Broadband
	5: Windows Media 8 Best Quality based VBR for Broadband
FileName	Saved file name

Return Values:

TRUE :	return "TRUE" if the record exists
FALSE:	return "TRUE" if the record doesn't exist

Save Image

void SaveImage(long iView ,LPCTSTR FileName);

Save the current image to jpg or bmp format

Notice: OpenRecordByDateTime() has to be called first

Parameters:

iCameraIndex	The selected camera index
FileName	Use the file extension to decide the image file format
	Support jpeg and bmp (*.jpg, *.bmp)

4.4 Display control and information feedback

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)

On Event

void OnEvent(long iEventID, long iParameter);

System event handler

Parameter:

iEventID:	0: OnConnect
	1: OnConnectFailed
	2: OnDisconnect
	3: OnOpenSuccess
iParameter:	?

On Display Time

void OnDisplayTime(BSTR DateTime);

Display time information.

Parameter:

DateTime:	YYYYMMDDHHMMSS