Playback ActiveX Control Properties

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

Playback ActiveX Control Method

BOOL Connect();

Connect to remote server

Return Values:

TRUE	Successfully to connect to server
FALSE	Failed to connect to server

BOOL Disconnect();

Disconnect from server

Return Values:

TRUE	Successfully disconnect from server
FALSE	Failed to disconnect from server

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 begin of the selected record video (YYYYMMDDHHMMSS)
EndDateTime	the end 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.

iView the displayed view.

Return Values:

TRUE: Successfully to open recorded video

FALSE: Failed to open recorded video

BOOL SetActiveView(long iView);

Focus on specified view, need to call OpenRecordByDateTime() first.

Parameters:

iView Focus on which camera

Return Values:

TRUE Camera connected to this view already.

FALSE Camera doesn't connect to this view.

long GetActiveView();

Get the focused view,

Return Values:

Active view index

void GetCameraIndexFromView(long iView);

Get camera's index from view

Parameters:

iView Which view is selected.

long GetCameraCount();

Get camera number

Return Values:

Camera total number

BSTR GetCameraName(long iCameraIndex);

Get the camera name

Parameters:

iCameraIndex Camera's Index

Return Values:

Camera name

BSTR GetCameraModel(long iCameraIndex);

Get the camera model

Parameters:

iCameraIndex Camera's Index

Return Values:

Camera model

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)

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

Save the selected time interval as AVI, need to call OpenRecordByDateTime() first.

Parameters:

iView	the selected camera index
StartDateTime	the begin of the selected record video (YYYYMMDDHHMMSS)
EndDateTime	the end 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:	Record exist
FALSE:	Record doesn't exist

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

Save the selected time interval as ASF, need to call OpenRecordByDateTime() first.

Parameters:

iView	the selected camera index
StartDateTime	the begin of the selected record video (YYYYMMDDHHMMSS)
EndDateTime	the end 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:	Record exist
FALSE:	Record doesn't exist

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

Save the selected time interval as Original AVI, need to call OpenRecordByDateTime() first.

Parameters:

iView	the selected camera index
StartDateTime	the begin of the selected record video (YYYYMMDDHHMMSS)
EndDateTime	the end 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:	Record exist
FALSE:	Record doesn't exist

void SaveImage(long iView ,LPCTSTR FileName);

Save current image, need to call OpenRecordByDateTime() first.

Parameters:

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

void Play();

Play recorded video from playback server

void Pause();

Pause recorded video from playback server

void Stop();

Stop the recorded video from playback server

void StepBackward();

Pause the recorded video and move the time backward

void StepForward ();

Pause the recorded video and move the time forward

BOOL Seek(LPCTSTR DateTime)

Move to specific time

Parameter:

DateTime	Date time of selected record (YYYYMMDDHHMMSS)

Return Values:

TRUE:	The time interval is valid.
FALSE:	The time interval is nod valid.

void SpeedUp();

Set the video's playing speed faster

void SpeedDown();

Set the video's playing speed slower

BSTR GetCurrDisplayTime();

Get the timestamp of the current display frame.

Return Values:

Timestamp (Unit: microsecond)

void ChangeResolution(long iWidth, long iHeight);

Change the video window size.

Parameters:

iWidth	video window width
iHeight	video window height

void Previous()

To see the previous one minute image of current time

void Next()

To see the next one minute image of current time

BSTR GetPlayerState()

Get current player status.

Return Values:

NotConnected:	Disconnect to server
Paused:	Paused
Running:	Running
Stopped:	Stopped

void GetRecordDateList(BSTR StartDateTime, BSTR EndDateTime)

Get list of date in which video(s) are recorded. The result will return by **OnRecordDate.**

Parameters:

StartDateTime	the begin of the query date (YYYYMMDDHHMMSS)

EndDateTime the end of the query date (YYYYMMDDHHMMSS)

void GetScheduleLogs(BSTR DateTime)

Get list of recording schedules in the specific date. The result will return by **OnScheduleLog.**

Parameters:

DateTime the query date (YYYYMMDDHHMMSS)

void GetRecordLogs(BSTR DateTime)

Get list of recorded videos in the specific date. The result will return by **OnRecordLog.**

Parameters:

DateTime the query date (YYYYMMDDHHMMSS)

void QueryEvents(BSTR StartDateTime, BSTR EndDateTime, BSTR DeviceType, long iCameraIndex, BSTR EventType)

Query the list of recording events during specific time range. The result will return by **OnEventLog.**

Parameters:

StartDateTime the begin of the query date (YYYYMMDDHHMMSS)

EndDateTime the end of the query date (YYYYMMDDHHMMSS)

DeviceType Reserved

iCameraIndex the index of query camera, set it to -1 for all camera

EventType the event type, set it to empty string for all event type. Possible value are:

"Np_EVENT_GENERAL_MOTION_1"

"Np_EVENT_GENERAL_MOTION_2"

"Np_EVENT_GENERAL_MOTION_3"

"Np_EVENT_GENERAL_MOTION_4"

"Np_EVENT_GENERAL_MOTION_5"

"Np_EVENT_FOREIGN_OBJECT"

"Np_EVENT_MISSING_OBJECT"

"Np_EVENT_FOCUS_LOST"

"Np_EVENT_CAMERA_OCCLUSION"

"Np_EVENT_GENERAL_MOTION_DEVICE"

"Np_EVENT_SIGNAL_LOST"

"Np_EVENT_MOTION_START"

"Np_EVENT_MOTION_STOP"

"Np_EVENT_CONNECTION_LOST"

"Np_EVENT_MANUAL_RECORD_MODE_START"

"Np_EVENT_MANUAL_RECORD_MODE_STOP"

"Np_EVENT_INPUT_OPENED"

"Np_EVENT_INPUT_CLOSED"

Playback ActiveX Control Events

void OnEvent(long iEventID, long iParameter);

System event handler

Parameter:

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

void OnDisplayTime(BSTR DateTime);

Display time information.

Parameter:

Date:	YYYYMMDDHHMMSS

void OnRecordDate(BSTR DateTime);

The result of **GetRecordDateList**

Parameter:

DateTime	(YYYYMMDDHHMMSS)

void OnScheduleLog (long iCameraIndex, BSTR Type, BSTR StartTime, BSTR EndTime);

The result of **GetScheduleLogs**

Parameter:

iCameraIndex	The camera index of the log
Type	Schedule type of the log. The possible value are:
	"ScheduleRecordOnly"
	"ScheduleMotionDetectOnly"
	"ScheduleRecordAndMotionDetect"
	"ScheduleRecordMovingOnly"
	"ScheduleRecordOnEvent"
	"ScheduleUndefined"
	"ScheduleRecordBoost"
StartTime	The start time of the log (YYYYMMDDHHMMSS)
EndTime	The end time of the log (YYYYMMDDHHMMSS)

void OnRecordLog (long iCameraIndex, BSTR StartTime, BSTR EndTime);

The result of **GetRecordLogs**

Parameter:

iCameraIndex	The camera index of the log
StartTime	The start time of the log (YYYYMMDDHHMMSS)
EndTime	The end time of the log (YYYYMMDDHHMMSS)

void OnEventLog(BSTR DateTime, long cameraIndex, BSTR EventType, BSTR auxiliaryCode, BSTR description, BSTR sourceName);

The result of **QueryEvents**

Parameter:

DateTime	Time when the event present. (YYYYMMDDHHMMSS)
iCameraIndex	The camera index of the event
EventType	Event type. Possible value are: "Np_EVENT_GENERAL_MOTION_1" "Np_EVENT_GENERAL_MOTION_2" "Np_EVENT_GENERAL_MOTION_3" "Np_EVENT_GENERAL_MOTION_4" "Np_EVENT_GENERAL_MOTION_5" "Np_EVENT_FOREIGN_OBJECT" "Np_EVENT_MISSING_OBJECT" "Np_EVENT_FOCUS_LOST" "Np_EVENT_CAMERA_OCCLUSION" "Np_EVENT_GENERAL_MOTION_DEVICE" "Np_EVENT_SIGNAL_LOST" "Np_EVENT_MOTION_START" "Np_EVENT_MOTION_STOP" "Np_EVENT_CONNECTION_LOST" "Np_EVENT_MANUAL_RECORD_MODE_START" "Np_EVENT_MANUAL_RECORD_MODE_STOP" "Np_EVENT_INPUT_OPENED" "Np_EVENT_INPUT_CLOSED"
auxiliaryCode	Reserved
description	The description of the event
sourceName	Reserved

void OnExportFinish (BSTR Status);

The event of export video finished

Parameter:

Status	SUCCESS: Success to export video
	FAILED: Failed to export video