SIP Endpoint SDK Developer's Guide

Video Support

3/14/2025

Video Support

This article discusses the video support that has been added to SIP Endpoint SDK. This support consists of:

- A new video control interface
- New video call control features that have been added to the existing call control interface

This article also outlines the requirements for processing High Definition video.

High Definition Video Requirements

> **Important**
> If you plan on supporting High Definition (HD) video, please take the following requirements into account.

Due to the complex processing required for High Definition video, all of the endpoints that are involved in a video conversation must run on computers that meet a certain minimum hardware performance level. As the actual CPU performance can no longer be accurately measured in MHz and since video processing performance depends on a wide variety of factors, Genesys recommends that you use the free benchmarking tool NovaBench to assess whether your hardware meets the requirements for successful HD video processing.

For 720p HD video, the **minimum** requirements are (in addition to what the camera manufacturer requires):

- A total NovaBench score of at least 500 (with a Graphic Test sub-score of at least 12), with recommended scores of at least 800 and 20, respectively
- 2 GB RAM
- 1 Mbps upload and download speed (for a total bandwidth of 2 Mpbs)

SIP Endpoint SDK for .NET does not currently support video resolutions higher than 720p.

Video Control

The `IVideoControl` interface is in the `Genesyslab.Sip.Endpoint.Provider.CP` namespace. It allows you to work with events and information related to:

- Video source

- Frame events

These features are discussed in the following sections.

Video Source

This SDK provides methods to get or set video source–related information:

Method	Description
bool HasCaptureDevice();	Returns true if the SDK has been configured to work with a camera.
String^ GetCurrentCaptureDeviceName();	Returns the names of the devices currently being used for video.
array<VideoDevice^>^ GetAllSystemCaptureDevices();	Returns the names of all available cameras known to the operating system.
GsStatus SetCaptureDevice(int deviceId);	Picks up a capture device by its device ID and gets its status if the device has been successfully reserved.
VideoCapability^ GetVideoCapability(int deviceId, int N);	Returns capture device video capability by device ID and capability order number.
int GetNumberVideoCapabilities(int deviceId);	Returns a number of supported capabilities by device ID.

Video Frame Events

The SDK also provides video frame–related events:

Method	Description
event EventHandler<EndpointEventArgs^>^ VideoFrameSizeReceived;	Provides notification about video frame size.
event EventHandler<EndpointEventArgs^>^ VideoFrameDelivered;	Provides notification that a video frame has been received with size shown by `VideoFrameSizeReceived`.

Video Call Control

SIP Endpoint SDK's `ICallControl` interface, which is in the `Genesyslab.Sip.Endpoint.Provider.Genesys` namespace, has methods to dial and answer a session with or without video:

Method	Description
void Dial(int connectionId, String^ destination, bool video, String^ data);	Starts dialing to place an outgoing call from the connection with `ID = connectionId` to `destination`. If `video=true`, the video is offered.
void Answer(int sessionId, bool video);	Answers an incoming call with `sessionId` and if

Method	Description
	`video=true`, the offered video is accepted.

Access to Raw Video Frames

SIP Endpoint SDK 8.1.2 for .NET provides access to raw video frames in BGR32 format.

The following API has been added:

```
public enum class VideoRenderFormat {
        i420     = 0,
        YV12     = 1,
        YUY2     = 2,
        UYVY     = 3,
        IYUV     = 4,
        ARGB     = 5,
        RGB24    = 6,
        RGB565   = 7,
        ARGB4444 = 8,
        ARGB1555 = 9,
        MJPEG    = 10,
        NV12     = 11,
        NV21     = 12,
        BGRA     = 13,
        Unknown  = 99
};
```

You can use these methods to add and remove local and remote videos:

```
// Start/Stop Local Video to be shown in the Windows Form with handle
GsStatus StartLocalVideo(VideoCapability^ inOut, IntPtr handle,
         unsigned zOrder, float left, float top, float right, float bottom);
GsStatus StopLocalVideo();

// Start/Stop Remote Video for sessionId to be shown in the Windows Form with handle
GsStatus StartRemoteVideo(int sessionId, IntPtr handle, unsigned zOrder,
         float left, float top, float right, float bottom);
GsStatus StopRemoteVideo(int sessionId);
```

The following methods may be used to add or remove the local or remote video renderer:

```
// Add/Remove External Local Video Renderer
void AddLocalVideoRenderer(VideoRenderFormat format);
void RemoveLocalVideoRenderer();

// Add/Remove External Remote Video Renderer
void AddRemoteVideoRenderer(VideoRenderFormat videoFormat, int sessionId);
void RemoveRemoteVideoRenderer(int sessionId);
```

Adding Video to an Audio-Only Call

In order to add video to an existing session, one of the supported video codecs should be specified in the **SipEndpoint.config** file:

```
<domain name="codecs">
```

```
. . .
        <section name="vp8">
          <setting name="payload_type" value="100"/>
        </section>
        <section name="h264">
          <setting name="payload_type" value="108"/>
          <setting name="fmtp" value="profile-level-id=420028"/>
        </section>
        <section name="vp9">
           <setting name="payload_type" value="101"/>
        </section>
      </domain>
```

If **auto_accept_video** is enabled and the other party adds video during an active audio call, video is automatically added to the endpoint. However, if **auto_accept_video** is not enabled and the other party adds video during an active audio call, then the user is prompted to either accept or reject the video.

API

The `ICallControl` Interface has been extended with these methods:

```
        GsStatus SendVideoOffer(int sessionId);
        GsStatus AnswerVideoOffer(int sessionId, bool accept);
        GsStatus RemoveVideoStream(int sessionId);
```

You can add video to audio-only calls using both built-in and external video frame renderers.

Application Notes

Local video should be started before you add the video. If the media offer is accepted at this point, the video becomes bidirectional. If there is a request to remove the video, the local video is disconnected from the session.

Audio-video scenarios

Video Offer Rejected By Remote Party

Local party	Remote party	API
Establish audio-only session		
Voice to video escalation		
Start local video		
Request to add video		`SendVideoOffer(sessionId)`
	Receive session status	`SessionStatus.MediaOffer`
	Reject video offer	`AnswerVideoOffer(sessionId, FALSE)`
Receive session status		SessionStatus.MediaRejected
Audio-only session established		

Video Offer Accepted By Remote Party

Local party	Remote party	API
Add video to audio-only call		
Voice to video escalation		
Start local video		`StartLocalVideo()`
Request to add video		`SendVideoOffer(sessionId)`
	Receive session status	`SessionStatus.MediaOffer`
	Accept video offer	`AnswerVideoOffer(sessionId, TRUE)`
Receive session status		`SessionStatus.MediaAccepted`
Show remote video	Show remote video	`StartRemoteVideo(sesionId)`
Bidirectional video session established		

Video Removed From The Call

Local party	Remote party	API
Existing bidirectional video session		
Remove video		`RemoveVideoStream(sessionId)`
Receive session status	Receive session status	`SessionStatus.MediaRejected`
Close remote video	Close remote video	`StopRemoteVideo(sesionId)`
Voice only session		