

InterpreterConnect

API Documentation

February 9, 2026



Introduction

This guide describes our client library, which allows developers to add real-time communication capabilities to their applications using a simple API that spans multiple platforms (Windows, Mac, Linux, iOS, Android) and languages (C, Objective-C, C#, Java, Javascript).

CONNECTOR

The videoconferencing API that handles all aspects of the media (audio/video) connection.

CONSTRUCTOR

Syntax

```
CreateVidyoConnector({viewId:String, viewStyle:String, remoteParticipants:Number, logFileFilter:String, logFileName:String, userData:Number})
```

Parameters

Name	Type	Description
viewId		A platform-specific view ID where the VidyoConnector's rendering window will be added as a child window. It will be used to render the preview and composite remote participants. When running custom layout on macOS, the core animation layer should be disabled on this view.
viewStyle	String	Type of the composite renderer which represents the visual style and behaviour.
remoteParticipants	Number	Number of remote participants to composite into the window. Setting the value to 0 (zero) will render the preview only.
logFileFilter	String	A space-separated (or comma-separated) sequence of names of log levels, each optionally followed by a category. Categories are separated from levels by the character '@'. Either a level or category may be specified as 'all' or '*' (interchangeably), meaning all levels or categories. A level specified without a category is equivalent to a level@*.

		<p>The levels are: 'fatal', 'error', 'warning', 'info', 'debug', 'sent', 'received', 'enter', and 'leave'.</p> <p>The level may be prefixed by '-' or '!' (interchangeably), meaning to remove the given level (or levels, see below) from the given category. The string 'none' is equivalent to '-all'.</p> <p>The first level in the list may be prefixed by '+'. If the first level begins with a '+' or '-/!', the string is used to modify the listener's existing levels and categories. Otherwise, unless the add parameter is true, the levels and categories specified in the string replace all existing levels and categories for the listener. The leading character '+' may be present before other levels, but is ignored.</p> <p>Unless they are preceded by '=', level names imply other levels as well. In particular, each of the levels 'fatal', 'error', 'warning', 'info', and 'debug' implies the levels higher than it, when specified positively; 'sent' and 'received' always imply each other; and 'enter' and 'leave' always imply each other.</p> <p>See VidyoConnectorGetLogCategories for determining the names and descriptions of registered categories.</p> <p>Level names are case-insensitive; category names are case-sensitive.</p> <p>Example: 'all' - all levels, all categories.</p> <p>Example: 'all -enter' - all categories, all levels but enter and leave.</p> <p>Example: '*@VidyoClient' - all levels for the VidyoClient category.</p> <p>Example: '-*@VidyoClient' - remove all levels from the VidyoClient category.</p> <p>Example: 'debug' - all categories, fatal, error, warning, info, and debug levels.</p> <p>Example: '=debug' - all categories, debug level only.</p> <p>Example: 'all -=enter' - all categories, all levels but enter (leave is still included).</p> <p>Example: 'error@VidyoClient' - fatal and error only for the VidyoClient category.</p> <p>Example: 'warning debug@VidyoClient' - fatal, error, and warning for all categories; additionally, info and debug for the VidyoClient category.</p>
logFileName	String	Full path to the file where the log should be stored; otherwise, NULL or empty string, in order to use the default OS-dependent writable path.
userData	Number	Arbitrary user data that can be retrieved later.

Returns

Type	Description
------	-------------

Object	The constructed object on success. NULL on failure.
--------	---

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

METHODS

VIDYOCONNECTOR : ASSIGNVIEWTOCOMPOSITERENDERER

Composites the preview and remote participants in a view.

Syntax

```
AssignViewToCompositeRenderer({viewId:String, viewStyle:String, remoteParticipants:Number})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the participant will be rendered.
viewStyle	String	Type of the composite renderer which represents the visual style and behaviour.
remoteParticipants	Number	Number of remote participants to composite into the window. Setting the value to 0 will render preview only.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : ASSIGNVIEWTOLOCALCAMERA

Assigns the view in which to show the preview from a local camera.

Syntax

```
AssignViewToLocalCamera({viewId:String, localCamera:Object, displayCropped:Boolean, allowZoom:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the participant will be rendered.
localCamera	Object	The VidyoLocalCamera that will be shown.
displayCropped	Boolean	Render the image as cropped. Otherwise letterbox.
allowZoom	Boolean	Allow zooming of the image with mouse or touch.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : ASSIGNVIEWTOLOCALMONITOR

Shows a VidyoLocalMonitor in a view.

Syntax

```
AssignViewToLocalMonitor({viewId:String, localMonitor:Object, displayCropped:Boolean, allowZoom:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the VidyoLocalMonitor will be rendered.
localMonitor	Object	The VidyoLocalMonitor that will be shown.
displayCropped	Boolean	Render the image as cropped. Otherwise letterbox.
allowZoom	Boolean	Allow zooming of the image with mouse or touch.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : ASSIGNVIEWTOLOCALWINDOWSHARE

Shows a VidyoLocalWindowShare in a view.

Syntax

```
AssignViewToLocalWindowShare({viewId:String, localWindowShare:Object, displayCropped:Boolean, allowZoom:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the VidyoLocalWindowShare will be rendered.
localWindowShare	Object	The VidyoLocalWindowShare that will be shown.
displayCropped	Boolean	Render the image as cropped. Otherwise letterbox.
allowZoom	Boolean	Allow zooming of the image with mouse or touch.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : ASSIGNVIEWTOREMOTECAMERA

Assigns the view in which to show the remote camera of one of the other participants.

Syntax

```
AssignViewToRemoteCamera({viewId:String, remoteCamera:Object, displayCropped:Boolean, allowZoom:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the participant will be rendered.
remoteCamera	Object	The VidyoRemoteCamera that will be shown.
displayCropped	Boolean	Render the image as cropped. Otherwise letterbox.
allowZoom	Boolean	Allow zooming of the image with mouse or touch.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : ASSIGNVIEWTOREMOTEWINDOWSHARE

Shows a VidyoRemoteWindowShare in a view.

Syntax

```
AssignViewToRemoteWindowShare({viewId:String, remoteWindowShare:Object, displayCropped:Boolean, allowZoom:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the VidyoRemoteWindowShare will be rendered.
remoteWindowShare	Object	The VidyoRemoteWindowShare that will be shown.
displayCropped	Boolean	Render the image as cropped. Otherwise letterbox.
allowZoom	Boolean	Allow zooming of the image with mouse or touch.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : ASSIGNVIEWTOVIRTUALVIDEOSOURCE

Shows a VidyoVirtualVideoSource in a view.

Syntax

```
AssignViewToVirtualVideoSource({viewId:String, virtualVideoSource:Object, displayCropped:Boolean, allowZoom:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID where the VidyoVirtualVideoSource will be rendered.
virtualVideoSource	Object	The VidyoVirtualVideoSource that will be shown.
displayCropped	Boolean	Render the image as cropped. Otherwise letterbox.
allowZoom	Boolean	Allow zooming of the image with mouse or touch.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : CONNECT

Connects to a live conference with the specified resourceId.

Syntax

```
Connect({host:String, token:String, displayName:String, resourceId:String,
onSuccess:function, onFailure:function, onDisconnected:function})
```

Parameters

Name	Type	Description
host	String	Host to which the connection will be made.
token	String	Token generated by the backend that contains the user ID and other metadata that will be used for authentication.
displayName	String	A display name that will be used in the UI.
resourceId	String	An alphanumeric string. This resourceId does not need to be pre-provisioned but in order to create a connection between multiple participants, the same resourceId has to be specified.
onSuccess	function	Callback that is triggered when the connection is successfully established.
onFailure	function	Callback that is triggered when the connection establishment failed.
onDisconnected	function	Callback that is triggered when an established connection is disconnected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONDISCONNECTED

Callback that is triggered when an established connection is disconnected.

Syntax

```
Void OnDisconnected(reasonObj)
```

Parameters

Name	Type	Description
reasonObj	String	undefined

VIDYOCONNECTOR : ONFAILURE

Callback that is triggered when the connection attempt has failed.

Syntax

```
Void OnFailure(reasonObj)
```

Parameters

Name	Type	Description
reasonObj	String	undefined

VIDYOCONNECTOR : ONSUCCESS

Callback that is triggered when the connection is successfully established.

Syntax

```
Void OnSuccess()
```

VIDYOCONNECTOR : CREATEVIRTUALVIDEOSOURCE

Create virtual video source. The user can create a maximum of 10 virtual video sources.

Syntax

```
CreateVirtualVideoSource({type:String, id:String, name:String})
```

Parameters

Name	Type	Description
type	String	Type of the virtual video source.
id	String	Unique ID.
name	String	Name of the virtual video source.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : CYCLECAMERA

Cycles camera selection to the next available camera.

Syntax

```
CycleCamera()
```

Returns

Type	Description
------	-------------

Boolean	LMI_TRUE on success, LMI_FALSE on failure.
---------	--

VIDYOCONNECTOR : CYCLEMICROPHONE

Cycles microphone selection to the next available microphone.

Syntax

```
CycleMicrophone()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : CYCLESPEAKER

Cycles speaker selection to the next available speaker.

Syntax

```
CycleSpeaker()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : DESTROYVIRTUALVIDEOSOURCE

Destroy virtual video source.

Syntax

```
DestroyVirtualVideoSource({virtualVideoSource:Object})
```

Parameters

Name	Type	Description
virtualVideoSource	Object	The virtual video source that will be destroyed.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : DISABLE

Disables connector object, releases all devices as precursor to destruct.

Syntax

```
Disable()
```

Returns

Type	Description
Void	

VIDYOCONNECTOR : DISABLEDEBUG

Syntax

```
DisableDebug()
```

Returns

Type	Description
Void	

VIDYOCONNECTOR : DISCONNECT

Disconnects a live conference. The onDisconnected callback will be called on successful execution.

Syntax

```
Disconnect()
```

Returns

Type	Description
Void	

VIDYOCONNECTOR : ENABLEDEBUG

Enabled debug mode which will start debug logging and enable statistics.

Syntax

```
EnableDebug({port:Number, logFilter:String})
```

Parameters

Name	Type	Description
port	Number	Local listening port for debugging.
logFilter	String	Log filter to use on the debug port.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : GETCPUTRADEOFFPROFILE

Get current CPU tradeoff profile setting.

Syntax

```
GetCpuTradeOffProfile()
```

Returns

Type	Description
	The current CPU tradeoff profile.

VIDYOCONNECTOR : GETMAXSENDBITRATE

Return the max send bit rate configured by application.

Syntax

```
GetMaxSendBitRate()
```

Returns

Type	Description
Number	the max send bit rate configured by application, in bps

VIDYOCONNECTOR : GETRENDEREROPTIONSFORVIEWID

Get the local renderer and conference tiles displaying options.

Syntax

```
GetRendererOptionsForViewId({viewId:String})
```

Parameters

Name	Type	Description
viewId		A platform-specific view ID.

Returns

Type	Description
String	Pointer to LmiString that contains the local renderer and conference tiles displaying options in JSON format.

VIDYOCONNECTOR : GETSTATE

Retrieves the state of the VidyoConnector.

Syntax

```
GetState()
```

Returns

Type	Description
	The state of the VidyoConnector.

VIDYOCONNECTOR : GETSTATSJSON

Retrieves VidyoConnector statistics in JSON format.

Syntax

```
GetStatsJson()
```

Returns

Type	Description
String	JSON encoded string on success, NULL on failure.

VIDYOCONNECTOR : GETVERSION

Retrieves version of the VidyoConnector library.

Syntax

```
GetVersion()
```

Returns

Type	Description
String	Version on success, NULL on failure.

VIDYOCONNECTOR : GETVERSIONWITHOUTBUILDNUMBER

Retrieves version of the VidyoConnector library without the build number.

Syntax

```
GetVersionWithoutBuildNumber()
```

Returns

Type	Description
String	Version on success, NULL on failure.

VIDYOCONNECTOR : HIDEVIEW

Hides a view which is used for custom layout of sources.

Syntax

```
HideView({viewId:String})
```

Parameters

Name	Type	Description
viewId		The view ID of the view to hide.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : REGISTERLOCALCAMERAEVENTLISTENER

Registers to get notified about local camera events.

Syntax

```
RegisterLocalCameraEventListener({onAdded:function, onRemoved:function, onSelected:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered for every new or existing camera added.
onRemoved	function	Callback that is triggered when the camera is removed.
onSelected	function	Callback that is triggered when a camera is selected for use.
onStateUpdated	function	Callback that is triggered when a camera state is updated.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALCAMERAADDED

Callback that is triggered each time a local camera becomes available to a VidyConnector.

Syntax

```
Void OnLocalCameraAdded(VidyoLocalCameraObj)
```

Parameters

Name	Type	Description
VidyoLocalCameraObj	Object	undefined

VIDYOCONNECTOR : ONLOCALCAMERAREMOVED

Callback that is triggered each time a local camera is no longer available to a VidyoConnector.

Syntax

```
Void OnLocalCameraRemoved(VidyoLocalCameraObj)
```

Parameters

Name	Type	Description
VidyoLocalCameraObj	Object	undefined

VIDYOCONNECTOR : ONLOCALCAMERASELECTED

Callback that is triggered each time there is a change in which local camera is selected for use.

Syntax

```
Void OnLocalCameraSelected(VidyoLocalCameraObj)
```

Parameters

Name	Type	Description
VidyoLocalCameraObj	Object	undefined

VIDYOCONNECTOR : ONLOCALCAMERASTATEUPDATED

Callback that is triggered each time there is a change in the state of the camera such as being suspended or disabled.

Syntax

```
Void OnLocalCameraStateUpdated(VidyoLocalCameraObj, stateObj)
```

Parameters

Name	Type	Description
VidyoLocalCameraObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERLOCALCAMERAFRAMELISTENER

Registers to get notified about camera frames.

Syntax

```
RegisterLocalCameraFrameListener({onFrame:function, localCamera:Object, width:Number, height:Number, frameInterval:Number})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a remote camera.
localCamera	Object	The VidyoLocalCamera for which to receive the frames.
width	Number	Approximate width of the requested frame.
height	Number	Approximate height of the requested frame.
frameInterval	Number	Approximate height of the requested frame.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALCAMERAFRAME

Callback that is triggered each time a local camera produces a video frame.

Syntax

```
Void OnLocalCameraFrame(VidyoLocalCameraObj, VidyoVideoFrameObj)
```

Parameters

Name	Type	Description
VidyoLocalCameraObj	Object	undefined
VidyoVideoFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERLOCALMICROPHONEENERGYLISTENER

Registers to get notified about microphone energies.

Syntax

```
RegisterLocalMicrophoneEnergyListener({onEnergy:function})
```

Parameters

Name	Type	Description
onEnergy	function	Callback that is triggered for every frame from a microphone that has energy.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALMICROPHONEENERGY

Callback that is triggered each time a local microphone produces an audio frame with energy present.

Syntax

```
Void OnLocalMicrophoneEnergy(VidyoLocalMicrophoneObj, audioEnergyObj)
```

Parameters

Name	Type	Description
VidyoLocalMicrophoneObj	Object	undefined
audioEnergyObj	Number	undefined

VIDYOCONNECTOR : REGISTERLOCALMICROPHONEEVENTLISTENER

Registers to get notified about local microphone events.

Syntax

```
RegisterLocalMicrophoneEventListener({onAdded:function, onRemoved:function, onSelected:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered for every new or existing microphone added.
onRemoved	function	Callback that is triggered when the microphone is removed.

onSelected	function	Callback that is triggered when a microphone is selected for use.
onStateUpdated	function	Callback that is triggered when a microphone state is updated.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALMICROPHONEADDED

Callback that is triggered each time a local microphone becomes available to a VidyoConnector.

Syntax

```
Void OnLocalMicrophoneAdded(VidyoLocalMicrophoneObj)
```

Parameters

Name	Type	Description
VidyoLocalMicrophoneObj	Object	undefined

VIDYOCONNECTOR : ONLOCALMICROPHONEREMOVED

Callback that is triggered each time a local microphone is no longer available to a VidyoConnector.

Syntax

```
Void OnLocalMicrophoneRemoved(VidyoLocalMicrophoneObj)
```

Parameters

Name	Type	Description
------	------	-------------

VidyoLocalMicrophoneObj	Object	undefined
-------------------------	--------	-----------

VIDYOCONNECTOR : ONLOCALMICROPHONESELECTED

Callback that is triggered each time there is a change in which local microphone is selected for use.

Syntax

```
Void OnLocalMicrophoneSelected(VidyoLocalMicrophoneObj)
```

Parameters

Name	Type	Description
VidyoLocalMicrophoneObj	Object	undefined

VIDYOCONNECTOR : ONLOCALMICROPHONESTATEUPDATED

Callback that is triggered each time there is a change in the state of the microphone such as being suspended or disabled.

Syntax

```
Void OnLocalMicrophoneStateUpdated(VidyoLocalMicrophoneObj, stateObj)
```

Parameters

Name	Type	Description
VidyoLocalMicrophoneObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERLOCALMICROPHONEFRAMELISTENER

Registers to get notified about microphone frames.

Syntax

```
RegisterLocalMicrophoneFrameListener({onFrame:function, localMicrophone:Object})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a microphone.
localMicrophone	Object	The VidyoLocalMicrophone for which to receive the frames.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALMICROPHONEFRAME

Callback that is triggered each time a local microphone produces an audio frame.

Syntax

```
Void OnLocalMicrophoneFrame(VidyoLocalMicrophoneObj, VidyoAudioFrameObj)
```

Parameters

Name	Type	Description
VidyoLocalMicrophoneObj	Object	undefined
VidyoAudioFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERLOCALMONITOREVENTLISTENER

Registers to get notified about local monitor events.

Syntax

```
RegisterLocalMonitorEventListener({onAdded:function, onRemoved:function,
onSelected:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered for every new or existing monitor added.
onRemoved	function	Callback that is triggered when the monitor is removed.
onSelected	function	Callback that is triggered when a monitor is selected for use.
onStateUpdated	function	Callback that is triggered when a monitor state is updated.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALMONITORADDED

Callback that is triggered each time a local monitor becomes available to a VidyoConnector.

Syntax

```
Void OnLocalMonitorAdded(VidyoLocalMonitorObj)
```

Parameters

Name	Type	Description
VidyoLocalMonitorObj	Object	undefined

VIDYOCONNECTOR : ONLOCALMONITORREMOVED

Callback that is triggered each time a local monitor is no longer available to a VidyoConnector.

Syntax

```
Void OnLocalMonitorRemoved(VidyoLocalMonitorObj)
```

Parameters

Name	Type	Description
VidyoLocalMonitorObj	Object	undefined

VIDYOCONNECTOR : ONLOCALMONITORSELECTED

Callback that is triggered each time there is a change in which local monitor is selected for sharing.

Syntax

```
Void OnLocalMonitorSelected(VidyoLocalMonitorObj)
```

Parameters

Name	Type	Description
VidyoLocalMonitorObj	Object	undefined

VIDYOCONNECTOR : ONLOCALMONITORSTATEUPDATED

Callback that is triggered each time there is a change in the state of the monitor such as being suspended or disabled.

Syntax

```
Void OnLocalMonitorStateUpdated(VidyoLocalMonitorObj, stateObj)
```

Parameters

Name	Type	Description
------	------	-------------

VidyoLocalMonitorObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERLOCALMONITORFRAMELISTENER

Registers to get notified about monitor frames.

Syntax

```
RegisterLocalMonitorFrameListener({onFrame:function, localMonitor:Object, width:Number, height:Number, frameInterval:Number})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a monitor.
localMonitor	Object	The VidyoLocalMonitor for which to receive the frames.
width	Number	Approximate width of the requested frame.
height	Number	Approximate height of the requested frame.
frameInterval	Number	Approximate height of the requested frame.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALMONITORFRAME

Callback that is triggered each time a local camera produces a video frame.

Syntax

```
Void OnLocalMonitorFrame(VidyoLocalMonitorObj, VidyoVideoFrameObj)
```

Parameters

Name	Type	Description
VidyoLocalMonitorObj	Object	undefined
VidyoVideoFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERLOCALSPEAKEREVENTLISTENER

Registers to get notified about local speaker events.

Syntax

```
RegisterLocalSpeakerEventListener({onAdded:function, onRemoved:function, onSelected:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered for every new or existing speaker added.
onRemoved	function	Callback that is triggered when the speaker is removed.
onSelected	function	Callback that is triggered when a speaker is selected for use.
onStateUpdated	function	Callback that is triggered when a speaker state is updated.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALSPEAKERADDED

Callback that is triggered each time a local speaker becomes available to a VidyConnector.

Syntax

```
Void OnLocalSpeakerAdded(VidyoLocalSpeakerObj)
```

Parameters

Name	Type	Description
VidyoLocalSpeakerObj	Object	undefined

VIDYOCONNECTOR : ONLOCALSPEAKERREMOVED

Callback that is triggered each time a local speaker is no longer available to a VidyConnector.

Syntax

```
Void OnLocalSpeakerRemoved(VidyoLocalSpeakerObj)
```

Parameters

Name	Type	Description
VidyoLocalSpeakerObj	Object	undefined

VIDYOCONNECTOR : ONLOCALSPEAKERSELECTED

Callback that is triggered each time there is a change in which local speaker is selected for use.

Syntax

```
Void OnLocalSpeakerSelected(VidyoLocalSpeakerObj)
```

Parameters

Name	Type	Description
------	------	-------------

VidyoLocalSpeakerObj	Object	undefined
----------------------	--------	-----------

VIDYOCONNECTOR : ONLOCALSPEAKERSTATEUPDATED

Callback that is triggered each time there is a change in the state of the speaker such as being suspended or disabled.

Syntax

```
Void OnLocalSpeakerStateUpdated(VidyoLocalSpeakerObj, stateObj)
```

Parameters

Name	Type	Description
VidyoLocalSpeakerObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERLOCALWINDOWSHAREEVENTLISTENER

Registers to get notified about local shareable-window events.

Syntax

```
RegisterLocalWindowShareEventListener({onAdded:function, onRemoved:function, onSelected:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered for every new or existing window added.
onRemoved	function	Callback that is triggered when the window is removed.
onSelected	function	Callback that is triggered when a window is selected for use.
onStateUpdated	function	Callback that is triggered when a window state is updated.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALWINDOWSHAREADDED

Callback that is triggered each time a local window becomes available for a VidyoConnector to share in a conference.

Syntax

```
Void OnLocalWindowShareAdded(VidyoLocalWindowShareObj)
```

Parameters

Name	Type	Description
VidyoLocalWindowShareObj	Object	undefined

VIDYOCONNECTOR : ONLOCALWINDOWSHAREREMOVED

Callback that is triggered each time a local window is no longer available for a VidyoConnector to share in a conference.

Syntax

```
Void OnLocalWindowShareRemoved(VidyoLocalWindowShareObj)
```

Parameters

Name	Type	Description
VidyoLocalWindowShareObj	Object	undefined

VIDYOCONNECTOR : ONLOCALWINDOWSHARESELECTED

Callback that is triggered each time there is a change in which local window is selected for sharing.

Syntax

```
Void OnLocalWindowShareSelected(VidyoLocalWindowShareObj)
```

Parameters

Name	Type	Description
VidyoLocalWindowShareObj	Object	undefined

VIDYOCONNECTOR : ONLOCALWINDOWSHARESTATEUPDATED

Callback that is triggered each time there is a change in the state of the window such as being suspended or disabled.

Syntax

```
Void OnLocalWindowShareStateUpdated(VidyoLocalWindowShareObj, stateObj)
```

Parameters

Name	Type	Description
VidyoLocalWindowShareObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERLOCALWINDOWSHAREFRAMELISTENER

Registers to get notified about window frames.

Syntax

```
RegisterLocalWindowShareFrameListener({onFrame:function, localWindowShare:Object, width:Number, height:Number, frameInterval:Number})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a window.
localWindowShare	Object	The VidyoLocalWindowShare for which to receive the frames.
width	Number	Approximate width of the requested frame.
height	Number	Approximate height of the requested frame.
frameInterval	Number	Approximate height of the requested frame.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOCALWINDOWSHAREFRAME

Callback that is triggered each time a local camera produces a video frame.

Syntax

```
Void OnLocalWindowShareFrame(VidyoLocalWindowShareObj, VidyoVideoFrameObj)
```

Parameters

Name	Type	Description
VidyoLocalWindowShareObj	Object	undefined
VidyoVideoFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERLOGEVENTLISTENER

Registers to get notified about log events.

Syntax

```
RegisterLogEventListener({onLog: function, filter: String})
```

Parameters

Name	Type	Description
onLog	function	Callback that is triggered for every new log record.
filter	String	<p>A space-separated (or comma-separated) sequence of names of log levels, each optionally followed by a category. Categories are separated from levels by the character '@'.</p> <p>Either a level or category may be specified as 'all' or '*' (interchangeably), meaning all levels or categories. A level specified without a category is equivalent to a level@*.</p> <p>The levels are: 'fatal', 'error', 'warning', 'info', 'debug', 'sent', 'received', 'enter', and 'leave'.</p> <p>The level may be prefixed by '-' or '!' (interchangeably), meaning to remove the given level (or levels, see below) from the given category. The string 'none' is equivalent to '-all'.</p> <p>The first level in the list may be prefixed by '+'. If the first level begins with a '+' or '-/!', the string is used to modify the listener's existing levels and categories. Otherwise, unless the <code>add</code> parameter is true, the levels and categories specified in the string replace all existing levels and categories for the listener. The leading character '+' may be present before other levels, but is ignored.</p> <p>Unless they are preceded by '=', level names imply other levels as well. In particular, each of the levels 'fatal', 'error', 'warning', 'info', and 'debug' implies the levels higher than it, when specified positively; 'sent' and 'received' always imply each other; and 'enter' and 'leave' always imply each other.</p> <p>See <code>VidyoConnectorGetLogCategories</code> for determining the names and descriptions of registered categories.</p> <p>Level names are case-insensitive; category names are case-sensitive.</p> <p>Example: 'all' - all levels, all categories.</p> <p>Example: 'all -enter' - all categories, all levels but enter and leave.</p> <p>Example: '*@VidyoClient' - all levels for the VidyoClient category.</p> <p>Example: '-*@VidyoClient' - remove all levels from the VidyoClient category.</p> <p>Example: 'debug' - all categories, fatal, error, warning, info, and debug levels.</p> <p>Example: '=debug' - all categories, debug level only.</p> <p>Example: 'all -=enter' - all categories, all levels but enter (leave is still included).</p>

	<p>Example: 'error@VidyoClient' - fatal and error only for the VidyoClient category.</p> <p>Example: 'warning debug@VidyoClient' - fatal, error, and warning for all categories; additionally, info and debug for the VidyoClient category.</p>
--	---

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONLOG

Callback that is triggered for every log record.

Syntax

```
Void OnLog(VidyoLogRecordObj)
```

Parameters

Name	Type	Description
VidyoLogRecordObj	Object	undefined

VIDYOCONNECTOR : REGISTERMESSAGEEVENTLISTENER

Registers to get notified about message events.

Syntax

```
RegisterMessageEventListener({onChatMessageReceived: function})
```

Parameters

Name	Type	Description
------	------	-------------

onChatMessageReceived	function	Callback that is triggered when a chat message is received from a participant.
-----------------------	----------	--

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONCHATMESSAGE RECEIVED

Callback that is triggered each time a chat message is received from one of the other participants.

Syntax

```
Void OnChatMessageReceived(VidyoParticipantObj, VidyoChatMessageObj)
```

Parameters

Name	Type	Description
VidyoParticipantObj	Object	undefined
VidyoChatMessageObj	Object	undefined

VIDYOCONNECTOR : REGISTERNETWORKINTERFACEEVENTLISTENER

Registers to get notified about network interface events.

Syntax

```
RegisterNetworkInterfaceEventListener({onAdded:function, onRemoved:function, onSelected:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
------	------	-------------

onAdded	function	Callback that is triggered for every new or existing network interface added.
onRemoved	function	Callback that is triggered when the network interface is removed.
onSelected	function	Callback that is triggered when a network interface is selected for use.
onStateUpdated	function	Callback that is triggered when a network interface state changes.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONNETWORKINTERFACEADDED

Callback that is triggered each time a network interface is available to a VidyoConnector.

Syntax

```
Void OnNetworkInterfaceAdded(VidyoNetworkInterfaceObj)
```

Parameters

Name	Type	Description
VidyoNetworkInterfaceObj	Object	undefined

VIDYOCONNECTOR : ONNETWORKINTERFACEREMOVED

Callback that is triggered each time a network interface is no longer available to a VidyoConnector.

Syntax

```
Void OnNetworkInterfaceRemoved(VidyoNetworkInterfaceObj)
```

Parameters

Name	Type	Description
VidyoNetworkInterfaceObj	Object	undefined

VIDYOCONNECTOR : ONNETWORKINTERFACESELECTED

Callback that is triggered each time there is a change in which network interface is selected for use.

Syntax

```
Void OnNetworkInterfaceSelected(VidyoNetworkInterfaceObj, transportTypeObj)
```

Parameters

Name	Type	Description
VidyoNetworkInterfaceObj	Object	undefined
transportTypeObj	String	undefined

VIDYOCONNECTOR : ONNETWORKINTERFACESTATEUPDATED

Callback that is triggered each time there is a change in the state of the network interface such as being up or down.

Syntax

```
Void OnNetworkInterfaceStateUpdated(VidyoNetworkInterfaceObj, stateObj)
```

Parameters

Name	Type	Description
VidyoNetworkInterfaceObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERPARTICIPANTEVENTLISTENER

Registers to get notified about participant events.

Syntax

```
RegisterParticipantEventListener({onJoined:function, onLeft:function,
onDynamicChanged:function, onLoudestChanged:function})
```

Parameters

Name	Type	Description
onJoined	function	Callback that is triggered when another participant joins a conference.
onLeft	function	Callback that is triggered when an existing participant leaves a conference.
onDynamicChanged	function	Callback that is triggered when the order of participants has changed, based on their importance according to active speech detection.
onLoudestChanged	function	Callback that is triggered when a new participant becomes the loudest, based on active speech detection.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONDYNAMICPARTICIPANTCHANGED

Callback that is triggered each time the order of dyanmic participants has changed, based on active speech detection.

Syntax

```
Void OnDynamicParticipantChanged(participantsObj)
```

Parameters

Name	Type	Description
participantsObj	Array	undefined

VIDYOCONNECTOR : ONLOUDESTPARTICIPANTCHANGED

Callback that is triggered each time one of the other participants becomes the loudest, based on active speech detection.

Syntax

```
Void OnLoudestParticipantChanged(VidyoParticipantObj, audioOnlyObj)
```

Parameters

Name	Type	Description
VidyoParticipantObj	Object	undefined
audioOnlyObj	Boolean	undefined

VIDYOCONNECTOR : ONPARTICIPANTJOINED

Callback that is triggered each time another participant joins a conference.

Syntax

```
Void OnParticipantJoined(VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : ONPARTICIPANTLEFT

Callback that is triggered each time an existing participant leaves a conference.

Syntax

```
Void OnParticipantLeft(VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : REGISTERRECORDERINCALEVENTLISTENER

Registers to get notified when a recorder joins or leaves the call.

Syntax

```
RegisterRecorderInCallEventListener({onRecorderInCallChanged:function})
```

Parameters

Name	Type	Description
onRecorderInCallChanged	function	Callback that is triggered when a recorder joins or leaves the call.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : RECORDERINCALL

Callback that is triggered each time a recorder joins or leaves the call.

Syntax

```
Void RecorderInCall(recorderInCallObj)
```

Parameters

Name	Type	Description
recorderInCallObj	Boolean	undefined

VIDYOCONNECTOR : REGISTERREMOTECAMERAEVENTLISTENER

Registers to get notified about remote camera events.

Syntax

```
RegisterRemoteCameraEventListener({onAdded:function, onRemoved:function, onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered each time one of the other participants adds a remote camera into a conference.
onRemoved	function	Callback that is triggered each time one of the other participants removes a remote camera from a conference.
onStateUpdated	function	Callback that is triggered each time one of the other participants remote camera changes state in a conference.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREMOTECAMERAADDED

Callback that is triggered each time one of the other participants adds a remote camera into a conference.

Syntax

```
Void OnRemoteCameraAdded(VidyoRemoteCameraObj, VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoRemoteCameraObj	Object	undefined
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : ONREMOTECAMERAREMOVED

Callback that is triggered each time one of the other participants removes a remote camera from a conference.

Syntax

```
Void OnRemoteCameraRemoved(VidyoRemoteCameraObj, VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoRemoteCameraObj	Object	undefined
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : ONREMOTECAMERASTATEUPDATED

Callback that is triggered each time one of the other participants remote camera changes state in a conference.

Syntax

```
Void OnRemoteCameraStateUpdated(VidyoRemoteCameraObj, VidyoParticipantObj, stateObj)
```

Parameters

Name	Type	Description
VidyoRemoteCameraObj	Object	undefined
VidyoParticipantObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERREMOTECAMERAFRAMELISTENER

Registers to get notified about camera frames from other participants.

Syntax

```
RegisterRemoteCameraFrameListener({onFrame:function, remoteCamera:Object, width:Number, height:Number, frameInterval:Number})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a remote camera.
remoteCamera	Object	The VidyoRemoteCamera for which to receive the frames.
width	Number	Approximate width of the requested frame.
height	Number	Approximate height of the requested frame.
frameInterval	Number	Approximate height of the requested frame.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREMOTECAMERAFRAME

Callback that is triggered each time a remote camera produces a video frame.

Syntax

```
Void OnRemoteCameraFrame(VidyoRemoteCameraObj, VidyoParticipantObj, VidyoVideoFrameObj)
```

Parameters

Name	Type	Description
VidyoRemoteCameraObj	Object	undefined
VidyoParticipantObj	Object	undefined
VidyoVideoFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERREMOTEMICROPHONEENERGYLISTENER

Registers to get notified about microphone energies from other participants.

Syntax

```
RegisterRemoteMicrophoneEnergyListener({onEnergy: function})
```

Parameters

Name	Type	Description
onEnergy	function	Callback that is triggered for every frame with energy from a remote microphone.

Returns

Type	Description

Boolean LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREMOTE MICROPHONE ENERGY

Callback that is triggered each time a remote microphone produces an frame with energy.

Syntax

```
Void
OnRemoteMicrophoneEnergy(VidyoRemoteMicrophoneObj, VidyoParticipantObj, audioEnergyObj)
```

Parameters

Name	Type	Description
VidyoRemoteMicrophoneObj	Object	undefined
VidyoParticipantObj	Object	undefined
audioEnergyObj	Number	undefined

VIDYOCONNECTOR : REGISTERREMOTE MICROPHONE EVENT LISTENER

Registers to get notified about remote microphone events.

Syntax

```
RegisterRemoteMicrophoneEventListener({onAdded: function, onRemoved: function,
onStateUpdated: function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered when another participant adds a microphone into a conference.

onRemoved	function	Callback that is triggered when another participant removes a microphone from a conference.
onStateUpdated	function	Callback that is triggered when another participant microphone changes state in a conference.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREOTEMICROPHONEADDED

Callback that is triggered each time one of the other participants adds a remote microphone into a conference.

Syntax

```
Void OnRemoteMicrophoneAdded(VidyoRemoteMicrophoneObj, VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoRemoteMicrophoneObj	Object	undefined
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : ONREOTEMICROPHONEREMOVED

Callback that is triggered each time one of the other participants removes a remote microphone from a conference.

Syntax

```
Void OnRemoteMicrophoneRemoved(VidyoRemoteMicrophoneObj, VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoRemoteMicrophoneObj	Object	undefined
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : ONREMOTEMICROPHONESTATEUPDATED

Callback that is triggered each time one of the other participants remote microphone changes state in a conference.

Syntax

```
Void  
OnRemoteMicrophoneStateUpdated(VidyoRemoteMicrophoneObj, VidyoParticipantObj, stateObj)
```

Parameters

Name	Type	Description
VidyoRemoteMicrophoneObj	Object	undefined
VidyoParticipantObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERREMOTEMICROPHONEFRAMELISTENER

Registers to get notified about microphone frames from other participants.

Syntax

```
RegisterRemoteMicrophoneFrameListener({onFrame: function, remoteMicrophone: Object})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a remote microphone.
remoteMicrophone	Object	The VidyoRemoteMicrophone for which to receive the frames.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREOTEMICROPHONEFRAME

Callback that is triggered each time a remote microphone produces an audio frame.

Syntax

```
Void
OnRemoteMicrophoneFrame(VidyoRemoteMicrophoneObj, VidyoParticipantObj, VidyoAudioFrameObj
)
```

Parameters

Name	Type	Description
VidyoRemoteMicrophoneObj	Object	undefined
VidyoParticipantObj	Object	undefined
VidyoAudioFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERREMOTEWINDOWSHAREEVENTLISTENER

Registers to get notified about remote shreable-window events.

Syntax

```
RegisterRemoteWindowShareEventListener({onAdded:function, onRemoved:function,
onStateUpdated:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered when another participant adds a window-share into a conference.
onRemoved	function	Callback that is triggered when another participant removes a window-share from a conference.
onStateUpdated	function	Callback that is triggered when another participant window-share changes state in a conference.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREMOTEWINDOWSHAREADDED

Callback that is triggered each time one of the other participants adds, or shares, a remote window into a conference.

Syntax

```
Void OnRemoteWindowShareAdded(VidyoRemoteWindowShareObj, VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoRemoteWindowShareObj	Object	undefined

VidyoParticipantObj	Object	undefined
---------------------	--------	-----------

VIDYOCONNECTOR : ONREMOTEWINDOWSHAREREMOVED

Callback that is triggered each time one of the other participants removes a remote window from a conference, or stops sharing it.

Syntax

```
Void OnRemoteWindowShareRemoved(VidyoRemoteWindowShareObj, VidyoParticipantObj)
```

Parameters

Name	Type	Description
VidyoRemoteWindowShareObj	Object	undefined
VidyoParticipantObj	Object	undefined

VIDYOCONNECTOR : ONREMOTEWINDOWSHARESTATEUPDATED

Callback that is triggered each time one of the other participants remote window changes state in a conference.

Syntax

```
Void OnRemoteWindowShareStateUpdated(VidyoRemoteWindowShareObj, VidyoParticipantObj, stateObj)
```

Parameters

Name	Type	Description
VidyoRemoteWindowShareObj	Object	undefined
VidyoParticipantObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERREMOTEWINDOWSHAREFRAMELISTENER

Registers to get notified about window-share frames from other participants.

Syntax

```
RegisterRemoteWindowShareFrameListener({onFrame:function, remoteWindowShare:Object, width:Number, height:Number, frameInterval:Number})
```

Parameters

Name	Type	Description
onFrame	function	Callback that is triggered for every frame from a remote window share.
remoteWindowShare	Object	The VidyoRemoteWindowShare for which to receive the frames.
width	Number	Approximate width of the requested frame.
height	Number	Approximate height of the requested frame.
frameInterval	Number	Approximate height of the requested frame.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONREMOTEWINDOWSHAREFRAME

Callback that is triggered each time a remote camera produces a video frame.

Syntax

```
Void  
OnRemoteWindowShareFrame(VidyoRemoteWindowShareObj, VidyoParticipantObj, VidyoVideoFrameObj)
```

Parameters

Name	Type	Description
VidyoRemoteWindowShareObj	Object	undefined
VidyoParticipantObj	Object	undefined
VidyoVideoFrameObj	Object	undefined

VIDYOCONNECTOR : REGISTERRESOURCEMANAGEREVENTLISTENER

Registers to get notified about resource-management events.

Syntax

```
RegisterResourceManagerEventListener({onAvailableResourcesChanged: function,
onMaxRemoteSourcesChanged: function})
```

Parameters

Name	Type	Description
onAvailableResourcesChanged	function	Callback that is triggered when local resource availability changes.
onMaxRemoteSourcesChanged	function	Callback that is triggered when the maximum number of decoded sources changes.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONAVAILABLERESOURCESCHANGED

Callback that is triggered each time the available local resources change.

Syntax

```
Void
OnAvailableResourcesChanged(cpuEncodeObj, cpuDecodeObj, bandwidthSendObj, bandwidthReceiveObj)
```

Parameters

Name	Type	Description
cpuEncodeObj	Number	undefined
cpuDecodeObj	Number	undefined
bandwidthSendObj	Number	undefined
bandwidthReceiveObj	Number	undefined

VIDYOCONNECTOR : ONMAXREMOTE SOURCESCHANGED

Callback that is triggered each time the local client's maximum decoding capability changes.

Syntax

```
Void OnMaxRemoteSourcesChanged(maxRemoteSourcesObj)
```

Parameters

Name	Type	Description
maxRemoteSourcesObj	Number	undefined

VIDYOCONNECTOR : REGISTERVIRTUALVIDEO SOURCEEVENTLISTENER

Registers to get notified about virtual video source events.

Syntax

```
RegisterVirtualVideoSourceEventListener({onAdded:function, onRemoved:function, onStateUpdated:function, onExternalMediaBufferReleased:function})
```

Parameters

Name	Type	Description
onAdded	function	Callback that is triggered each time when virtual video source is added into a conference.
onRemoved	function	Callback that is triggered each time when virtual video source is added from a conference.
onStateUpdated	function	Callback that is triggered each time when virtual video source changes state in a conference.
onExternalMediaBufferReleased	function	Callback that is triggered each time when VidyoClient is done with an application allocated buffer that holds raw video frame.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONVIRTUALVIDEOADDED

Callback that is triggered each time a virtual video source becomes available for a VidyoConnector to add into a conference.

Syntax

```
Void OnVirtualVideoSourceAdded(VidyoVirtualVideoSourceObj)
```

Parameters

Name	Type	Description
VidyoVirtualVideoSourceObj	Object	undefined

VIDYOCONNECTOR :

ONVIRTUALVIDEOEXTERNALMEDIABUFFERRELEASED

Callback that is triggered to notify application that VidyoClient is done with the buffer that holds raw video frame. This buffer is the same one that was passed to VidyoClient via function VidyoVirtualVideoSourceSendFrameWithExternalData.

Syntax

```
Void OnVirtualVideoSourceExternalMediaBufferReleased(VidyoVirtualVideoSourceObj, bufferObj, sizeObj)
```

Parameters

Name	Type	Description
VidyoVirtualVideoSourceObj	Object	undefined
bufferObj	Number	undefined
sizeObj	Number	undefined

VIDYOCONNECTOR : ONVIRTUALVIDEOSOURCEREMOVED

Callback that is triggered each time a virtual video source is no longer available for a VidyoConnector to add into a conference.

Syntax

```
Void OnVirtualVideoSourceRemoved(VidyoVirtualVideoSourceObj)
```

Parameters

Name	Type	Description
VidyoVirtualVideoSourceObj	Object	undefined

VIDYOCONNECTOR : ONVIRTUALVIDEOSTATEUPDATED

Callback that is triggered each time there is a change in the state of the window such as being suspended or disabled.

Syntax

```
Void OnVirtualVideoSourceStateUpdated(VidyoVirtualVideoSourceObj, stateObj)
```

Parameters

Name	Type	Description
VidyoVirtualVideoSourceObj	Object	undefined
stateObj	String	undefined

VIDYOCONNECTOR : REGISTERWEBPROXYEVENTLISTENER

Registers to get notified about web proxy events.

Syntax

```
RegisterWebProxyEventListener({onWebProxyCredentialsRequest:function})
```

Parameters

Name	Type	Description
onWebProxyCredentialsRequest	function	Callback that is triggered when web proxy credentials are requested.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

CALLBACKS

VIDYOCONNECTOR : ONWEBPROXYCREDENTIALSREQUEST

Callback that is triggered when Web Proxy Server requires password.

Syntax

```
Void OnWebProxyCredentialsRequest(webProxyAddressObj)
```

Parameters

Name	Type	Description
webProxyAddressObj	String	undefined

VIDYOCONNECTOR : REPORTLOCALPARTICIPANTONJOINED

By default, local participant joining/leaving are not reported to applicaiotn. Use this function to report those events.

Syntax

```
ReportLocalParticipantOnJoined({reportLocalParticipant: Boolean})
```

Parameters

Name	Type	Description
reportLocalParticipant	Boolean	Set to LMI_TRUE if application is interested in local participant to be included in participant event callbacks. Otherwise set to LMI_FALSE. Default is LMI_FALSE

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTAUDIOCONTENTSHARE

Selects the local microphone to use for audio content share.

Syntax

```
SelectAudioContentShare({localMicrophone:Object})
```

Parameters

Name	Type	Description
localMicrophone	Object	Local microphone to select for audio content share. NULL if none of them should be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTDEFAULTCAMERA

Resets the camera selection to pick the Default camera.

Syntax

```
SelectDefaultCamera()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTDEFAULTMICROPHONE

Resets the microphone selection to pick the Default microphone.

Syntax

```
SelectDefaultMicrophone()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTDEFAULTNETWORKINTERFACEFORMEDIA

Selects the default ANY network interface to be used for Media.

Syntax

```
SelectDefaultNetworkInterfaceForMedia()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTDEFAULTNETWORKINTERFACEFORSIGNALING

Selects the default ANY network interface to be used for Signaling.

Syntax

```
SelectDefaultNetworkInterfaceForSignaling()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTDEFAULTSPEAKER

Resets the speaker selection to pick the Default speaker.

Syntax

```
SelectDefaultSpeaker()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTLOCALCAMERA

Selects the local camera to use.

Syntax

```
SelectLocalCamera({localCamera:Object})
```

Parameters

Name	Type	Description
localCamera	Object	Local camera to select. NULL if none of them should be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTLOCALMICROPHONE

Selects the local microphone to use.

Syntax

```
SelectLocalMicrophone({localMicrophone:Object})
```

Parameters

Name	Type	Description
localMicrophone	Object	Local microphone to select. NULL if none of them should be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTLOCALMONITOR

Selects the local monitor to share in a conference.

Syntax

```
SelectLocalMonitor({localMonitor:Object})
```

Parameters

Name	Type	Description
localMonitor	Object	Local monitor to share in a conference. NULL if none of them should be shared.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTLOCALSPEAKER

Selects the local speaker to use.

Syntax

```
SelectLocalSpeaker({localSpeaker:Object})
```

Parameters

Name	Type	Description
localSpeaker	Object	Local speaker to select. NULL if none of them should be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTLOCALWINDOWSHARE

Selects the local window to share in a conference.

Syntax

```
SelectLocalWindowShare({localWindowShare:Object})
```

Parameters

Name	Type	Description
localWindowShare	Object	Local window to share in a conference. NULL if none of them should be shared.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTNETWORKINTERFACEFORMEDIA

Selects the network interface to be used for Media.

Syntax

```
SelectNetworkInterfaceForMedia({networkInterface:Object})
```

Parameters

Name	Type	Description
networkInterface	Object	Network interface to be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTNETWORKINTERFACEFOR SIGNALING

Selects the network interface to be used for Signaling.

Syntax

```
SelectNetworkInterfaceForSignaling({networkInterface:Object})
```

Parameters

Name	Type	Description
networkInterface	Object	Network interface to be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTVIDEOCONTENTSHARE

Selects the local camera to use for video content share.

Syntax

```
SelectVideoContentShare({localCamera:Object})
```

Parameters

Name	Type	Description
localCamera	Object	Local camera to select. NULL if none of them should be selected.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTVIRTUALCAMERA

Selects the virtual video source which is camera that will be shown to remote participants. The real camera will be unselected if it's selected.

Syntax

```
SelectVirtualCamera({virtualCamera:Object})
```

Parameters

Name	Type	Description
virtualCamera	Object	The virtual video source that will be shown to remote participants.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SELECTVIRTUALWINDOWSHARE

Selects the virtual video source which is window share that will be shown to remote participants. The local window share will be unselected if it's selected.

Syntax

```
SelectVirtualWindowShare({virtualWindowShare:Object})
```

Parameters

Name	Type	Description
virtualWindowShare	Object	The virtual video source that will be shown to remote participants.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SENDCHATMESSAGE

Sends a message to all the other participants.

Syntax

```
SendChatMessage({message:String})
```

Parameters

Name	Type	Description
message	String	A message to send to all the other participants.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETADVANCEDOPTIONS

Sets up advanced options on the VidyoConnector API.

Syntax

```
SetAdvancedOptions({options:String})
```

Parameters

Name	Type	Description
options	String	The experimental options in JSON format.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETCAMERANUDGECONFIGURATION

Sets camera PTZ nudge configuration.

Syntax

```
SetCameraNudgeConfiguration({cameraNudgeConfig:String})
```

Parameters

Name	Type	Description
cameraNudgeConfig	String	Camera PTZ nudge configuration in JSON format.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETCAMERAPRIVACY

Sets the flag to show/hide the local camera for remote participants for composite/custom layout and local camera preview for composite layout only.

Syntax

```
SetCameraPrivacy({privacy: Boolean})
```

Parameters

Name	Type	Description
privacy	Boolean	Flag to show/hide the local camera.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETCERTIFICATEAUTHORITYLIST

Sets the list of certificate authority's certificates used to authenticate a TLS connection.

Syntax

```
SetCertificateAuthorityList({certificateAuthorityList:String})
```

Parameters

Name	Type	Description
certificateAuthorityList	String	The string representing Certificate Authority lists. Note: If the string 'default' is passed here, VidyoConnector will use a built-in Certificate Authority list.

Returns

Type	Description
Void	

VIDYOCONNECTOR : SETCPUOFFPROFILE

Set CPU tradeoff profile.

Syntax

```
SetCpuTradeOffProfile({profile:String})
```

Parameters

Name	Type	Description
profile	String	The tradeoff profile.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETFONTFILENAME

Set font file name.

Syntax

```
SetFontFileName({fontFileName:String})
```

Parameters

Name	Type	Description
fontFileName	String	Pointer to char array that contains font file name.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETLOCATION

Sets the geographic location of the endpoint using the VidyConnector.

Syntax

```
SetLocation({latitude: Number, longitude: Number})
```

Parameters

Name	Type	Description
latitude	Number	Latitude of the geographic coordinate.
longitude	Number	Longitude of the geographic coordinate.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETMAXBITRATE

Sets the max send bit rate for all sources.

Syntax

```
SetMaxBitRate({bitRate: Number})
```

Parameters

Name	Type	Description
bitRate	Number	The max send bit rate for all sources.

Returns

Type	Description
------	-------------

Void	
------	--

VIDYOCONNECTOR : SETMAXSENDBITRATE

Sets the max send bit rate for all sources.

Syntax

```
SetMaxSendBitRate({bitRate: Number})
```

Parameters

Name	Type	Description
bitRate	Number	The max send bit rate for all sources, in bps.

Returns

Type	Description
Void	

VIDYOCONNECTOR : SETMICROPHONEPRIVACY

Sets the flag to mute/unmute the microphone.

Syntax

```
SetMicrophonePrivacy({privacy: Boolean})
```

Parameters

Name	Type	Description
privacy	Boolean	Flag to mute/unmute the microphone.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETMODE

Sets the operating mode of the VidyoConnector. This API have to be called before renderer is destructed: i.e. VidyoConnectorHideView will be called in case VIDYO_CONNECTORMODE_Background and after renderer is constructed: i.e. VidyoConnectorAssignViewToCompositeRenderer, VidyoConnectorAssignViewToLocalCamera, VidyoConnectorAssignViewToRemoteCamera will be called in case VIDYO_CONNECTORMODE_Foreground. This API is required only when user in the conference.

Syntax

```
SetMode({mode:String})
```

Parameters

Name	Type	Description
mode	String	Desired operating mode.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETPOOL

Sets a specific media pool to use for audio and video.

Syntax

```
SetPool({name:String})
```

Parameters

Name	Type	Description
------	------	-------------

name	String	Name of the pool.
------	--------	-------------------

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETRENDEREROPTIONSFORVIEWID

Sets up the local renderer and conference tiles displaying options.

Syntax

```
SetRendererOptionsForViewId({viewId:String, options:String})
```

Parameters

Name	Type	Description
viewId		A platform-specific view ID.
options	String	A char array that contains options in the JSON format Keys and values are case-sensitive. List of supported keys and their values: Key: 'EnableVerticalVideoCentering'; Values: 'true', 'false'. Key: 'EnableFECCIconCustomLayout'; Values: 'true', 'false'. Example: '{'EnableVerticalVideoCentering':true, 'EnableFECCIconCustomLayout':false}'.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETSPEAKERPRIVACY

Sets the flag to mute/unmute the speaker.

Syntax

```
SetSpeakerPrivacy({privacy: Boolean})
```

Parameters

Name	Type	Description
privacy	Boolean	Flag to mute/unmute the speaker.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETTCPTRANSPORT

Enable the use of TCP Transport when connecting.

Syntax

```
SetTCPTransport({enable: Boolean})
```

Parameters

Name	Type	Description
enable	Boolean	Enable the transport.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETTURNADDRESSCREDENTIALS

Set TURN server and credentials information.

Syntax

```
SetTURNAddressCredentials({turnAddress:String, port:Number, username:String, password:String, baseTransport:String})
```

Parameters

Name	Type	Description
turnAddress	String	TURN server FQDN or IP address.
port	Number	TURN server port.
username	String	TURN server user name.
password	String	TURN server password.
baseTransport	String	Base transport type.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETTURNSEVERCERTIFICATEREQUIRED

Set TURN server certificate requirement.

Syntax

```
SetTURNServerCertificateRequired({required:Boolean})
```

Parameters

Name	Type	Description
required	Boolean	LMI_TRUE if TURN server certificate is required.

RETURNS

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETUDPTransport

Enable the use of UDP Transport when connecting.

Syntax

```
SetUDPTransport({enable: Boolean})
```

Parameters

Name	Type	Description
enable	Boolean	Enable the transport.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETVIEWAnimationSpeed

Sets the speed of transition animation for windows moving around inside the view.

Syntax

```
SetViewAnimationSpeed({viewId: String, speedPercentage: Number})
```

Parameters

Name	Type	Description
------	------	-------------

viewId		A platform specific view ID to be positioned.
speedPercentage	Number	Percentage of animation speed with default of 50, 100 for slowest and 0 for instant.

Returns

Type	Description
Boolean	

VIDYOCONNECTOR : SETVIEWBACKGROUNDCOLOR

Set the background color of the view.

Syntax

```
SetViewBackgroundColor({viewId:String, red:Number, green:Number, blue:Number})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID to be positioned.
red	Number	Red component of the color 0-255.
green	Number	Green component of the color 0-255.
blue	Number	Blue component of the color 0-255.

Returns

Type	Description
Boolean	

VIDYOCONNECTOR : SETWEBPROXYADDRESSCREDENTIALS

Sets web proxy server address and credentials.

Syntax

```
SetWebProxyAddressCredentials({proxyAddress:String, username:String, password:String})
```

Parameters

Name	Type	Description
proxyAddress	String	The web proxy server address.
username	String	The web proxy server user name.
password	String	The web proxy server password.

Returns

Type	Description
Void	

VIDYOCONNECTOR : SETWEBPROXYTRANSPORT

Enable the use of WebProxy Transport when connecting.

Syntax

```
SetWebProxyTransport({enable:Boolean})
```

Parameters

Name	Type	Description
enable	Boolean	Enable the transport.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SETWEBPROXYTRANSPORTADDRESS

Enable the use of WebProxy Transport with provided web proxy server address.

Syntax

```
SetWebProxyTransportAddress({name:String, port:Number})
```

Parameters

Name	Type	Description
name	String	The name of web proxy server, it may also may be prefixed by a scheme ('http://proxy.company.com')
port	Number	The port that web proxy server is running on

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SHOWAUDIOMETERS

Shows or hides the audio meters in the view.

Syntax

```
ShowAudioMeters({viewId:String, showMeters:Boolean})
```

Parameters

Name	Type	Description
------	------	-------------

viewId		A platform specific view ID to be positioned.
showMeters	Boolean	Boolean to show or hide the audio meters.

Returns

Type	Description
Boolean	

VIDYOCONNECTOR : SHOWPREVIEW

Sets the flag to show/hide the local camera preview for composite layout only.

Syntax

```
ShowPreview({preview:Boolean})
```

Parameters

Name	Type	Description
preview	Boolean	Flag to show/hide the local camera preview.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SHOWVIEWAT

Sets the position of the view every time it's changed on the screen.

Syntax

```
ShowViewAt({viewId:String, x:Number, y:Number, width:Number, height:Number})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID to be positioned.
x	Number	X coordinate.
y	Number	Y coordinate.
width	Number	Width of the view.
height	Number	Height of the view.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SHOWVIEWATPOINTS

Sets the position of the view every time it's changed on the screen.

Syntax

```
ShowViewAtPoints({viewId:String, x:Number, y:Number, width:Number, height:Number})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID to be positioned.
x	Number	X coordinate (in points).
y	Number	Y coordinate (in points).
width	Number	Width of the view (in points).

height	Number	Height of the view (in points).
--------	--------	---------------------------------

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : SHOWVIEWLABEL

Shows or hides the label on the view which displays the participant name or preview.

Syntax

```
ShowViewLabel({viewId:String, showLabel:Boolean})
```

Parameters

Name	Type	Description
viewId		A platform specific view ID to be positioned.
showLabel	Boolean	Boolean to show or hide the label.

Returns

Type	Description
Boolean	

VIDYOCONNECTOR : SHOWWINDOWSHAREPREVIEW

Sets the flag to show/hide the window share preview.

Syntax

```
ShowWindowSharePreview({preview:Boolean})
```

Parameters

Name	Type	Description
preview	Boolean	Flag to show/hide the window share preview.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALCAMERAEVENTLISTENER

Unregisters local camera event notifications.

Syntax

```
UnregisterLocalCameraEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALCAMERAFRAMELISTENER

Unregisters local camera frame notifications.

Syntax

```
UnregisterLocalCameraFrameListener({localCamera:Object})
```

Parameters

Name	Type	Description
localCamera	Object	The VidyoLocalCamera for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALMICROPHONEENERGYLISTENER

Unregisters local microphone energy notifications.

Syntax

```
UnregisterLocalMicrophoneEnergyListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALMICROPHONEEVENTLISTENER

Unregisters local microphone event notifications.

Syntax

```
UnregisterLocalMicrophoneEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALMICROPHONEFRAMELISTENER

Unregisters local microphone frame notifications.

Syntax

```
UnregisterLocalMicrophoneFrameListener({localMicrophone:Object})
```

Parameters

Name	Type	Description
localMicrophone	Object	The VidyLocalMicrophone for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALMONITOREVENTLISTENER

Unregisters local monitor event notifications.

Syntax

```
UnregisterLocalMonitorEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALMONITORFRAMELISTENER

Unregisters local monitor frame notifications.

Syntax

```
UnregisterLocalMonitorFrameListener({localMonitor:Object})
```

Parameters

Name	Type	Description
localMonitor	Object	The VidyoLocalMonitor for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALSPEAKEREVENTLISTENER

Unregisters local speaker event notifications.

Syntax

```
UnregisterLocalSpeakerEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALWINDOWSHAREEVENTLISTENER

Unregisters local window share event notifications.

Syntax

```
UnregisterLocalWindowShareEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOCALWINDOWSHAREFRAMELISTENER

Unregisters local window share frame notifications.

Syntax

```
UnregisterLocalWindowShareFrameListener({localWindowShare:Object})
```

Parameters

Name	Type	Description
localWindowShare	Object	The VidyLocalWindowShare for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERLOGEVENTLISTENER

Unregisters log event notifications.

Syntax

```
UnregisterLogEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERMESSAGEEVENTLISTENER

Unregisters message event notifications.

Syntax

```
UnregisterMessageEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERNETWORKINTERFACEEVENTLISTENER

Unregisters network interface event notifications.

Syntax

```
UnregisterNetworkInterfaceEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERPARTICIPANTEVENTLISTENER

Unregisters participant event notifications.

Syntax

```
UnregisterParticipantEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERRECORDERINCALEVENTLISTENER

Unregisters recorder in call event notifications.

Syntax

```
UnregisterRecorderInCallEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTECAMERAEVENTLISTENER

Unregisters remote camera event notifications.

Syntax

```
UnregisterRemoteCameraEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTECAMERAFRAMELISTENER

Unregisters remote camera frame notifications.

Syntax

```
UnregisterRemoteCameraFrameListener({remoteCamera:Object})
```

Parameters

Name	Type	Description
remoteCamera	Object	The VidyoRemoteCamera for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTEMICROPHONEENERGYLISTENER

Unregisters remote microphone frame notifications.

Syntax

```
UnregisterRemoteMicrophoneEnergyListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTEMICROPHONEEVENTLISTENER

Unregisters remote microphone event notifications.

Syntax

```
UnregisterRemoteMicrophoneEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTEMICROPHONEFRAMELISTENER

Unregisters remote microphone frame notifications.

Syntax

```
UnregisterRemoteMicrophoneFrameListener({remoteMicrophone:Object})
```

Parameters

Name	Type	Description
remoteMicrophone	Object	The VidyoRemoteMicrophone for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTEWINDOWSHAREEVENTLISTENER

Unregisters remote window share event notifications.

Syntax

```
UnregisterRemoteWindowShareEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERREMOTEWINDOWSHAREFRAMELISTENER

Unregisters remote window share frame notifications.

Syntax

```
UnregisterRemoteWindowShareFrameListener({remoteWindowShare:Object})
```

Parameters

Name	Type	Description
remoteWindowShare	Object	The VidyoRemoteWindowShare for which to stop notifications.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERRESOURCEMANAGEREVENTLISTENER

Unregisters resource manager event notifications.

Syntax

```
UnregisterResourceManagerEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERVIRTUALVIDEOSOURCEEVENTLISTENER

Unregisters virtual video source event notifications.

Syntax

```
UnregisterVirtualVideoSourceEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

VIDYOCONNECTOR : UNREGISTERWEBPROXYEVENTLISTENER

Unregisters web proxy event notifications.

Syntax

```
UnregisterWebProxyEventListener()
```

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE on failure.

AUDIOFRAME

This object represents an audio frame.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
bitsPerSample	Number	Bits per sample of the audio data which is based on the data format (16 bits for in16, 32 bits for f132).
data	Number	Pointer to the frame's payload.

discontinuity	Boolean	Discontinuity has been detected.
discontinuityBitPresent	Boolean	Discontinuity bit has been set.
elapsedTime	Number	Elapsed time since the beginning of frame capture that is increased by the duration of each frame.
energy	Number	Energy of the audio frame in dBFS (full scale).
energyBitPresent	Boolean	Energy bit has been set.
format	String	Format of the audio frame.
numberOfChannels	Number	Number of channels.
numberOfSamples	Number	Number of samples.
sampleRate	Number	Number of samples per second.
size	Number	Size in bytes of the audio frame object's payload.
speech	Boolean	Speech is detected in the frame.
speechBitPresent	Boolean	Speech bit has been set.
timestamp	Number	Wallclock timestamp associated with the creation of the audio frame object.

METHODS

VIDYOAUDIOFRAME : GETBITSPERSAMPLE

Syntax

```
GetBitsPerSample()
```

Returns

Type	Description
------	-------------

Number	.
--------	---

VIDYOAUDIOFRAME : GETDATA

Syntax

```
GetData()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETDISCONTINUITY

Syntax

```
GetDiscontinuity()
```

Returns

Type	Description
Boolean	.

VIDYOAUDIOFRAME : GETDISCONTINUITYBITPRESENT

Syntax

```
GetDiscontinuityBitPresent()
```

Returns

Type	Description
------	-------------

Boolean	.
---------	---

VIDYOAUDIOFRAME : GETELAPSEDTIME

Syntax

```
GetElapsedTime()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETENERGY

Syntax

```
GetEnergy()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETENERGYBITPRESENT

Syntax

```
GetEnergyBitPresent()
```

Returns

Type	Description
------	-------------

Boolean	.
---------	---

VIDYOAUDIOFRAME : GETFORMAT

Gets the frame format.

Syntax

```
GetFormat()
```

Returns

Type	Description
String	Frame format.

VIDYOAUDIOFRAME : GETNUMBEROFCHANNELS

Syntax

```
GetNumberOfChannels()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETNUMBEROFSAMPLES

Syntax

```
GetNumberOfSamples()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETSAMPLERATE

Syntax

```
GetSampleRate()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETSIZE

Syntax

```
GetSize()
```

Returns

Type	Description
Number	.

VIDYOAUDIOFRAME : GETSPEECH

Syntax

```
GetSpeech()
```

Returns

Type	Description
Boolean	.

VIDYOAUDIOFRAME : GETSPEECHBITPRESENT

Syntax

```
GetSpeechBitPresent()
```

Returns

Type	Description
Boolean	.

VIDYOAUDIOFRAME : GETTIMESTAMP

Syntax

```
GetTimestamp()
```

Returns

Type	Description
Number	.

DEVICE

A wrapper class for Local and Remote devices.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID of the device.
name	String	Name of the device.
type	String	Type of the device.

METHODS

VIDYODEVICE : GETID

Gets the device id.

Syntax

```
GetId()
```

Returns

Type	Description
String	Device id.

VIDYODEVICE : GETNAME

Gets the device name.

Syntax

```
GetName()
```

Returns

Type	Description
String	Device name.

VIDYODEVICE : GETTYPE

Gets the device type.

Syntax

```
GetType()
```

Returns

Type	Description
	Device Type.

LOCALCAMERA

This object represents a camera on the local endpoint.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.

METHODS

VIDYOLOCALCAMERA : ALLOWREMOTECAMERACONTROL

Allows the PTZ control of the camera by remote participants.

Syntax

```
AllowRemoteCameraControl({allow:Boolean})
```

Parameters

Name	Type	Description
allow	Boolean	Allow or prohibit camera control.

Returns

Type	Description
Boolean	LMI_TRUE on success.

VIDYOLOCALCAMERA : CONTROLPTZ

Send the PTZ Nudge commands to the local camera.

Syntax

```
ControlPTZ({pan:Number, tilt:Number, zoom:Number})
```

Parameters

Name	Type	Description
pan	Number	Number of nudges to pan the camera.
tilt	Number	Number of nudges to tilt the camera.
zoom	Number	Number of nudges to zoom the camera.

Returns

Type	Description
Boolean	LMI_TRUE when the control command was sent, LMI_FALSE otherwise.

VIDYOLOCALCAMERA : GETBACKLIGHTCOMPENSATION

Gets the backlight compensation state.

Syntax

```
GetBacklightCompensation()
```

Returns

Type	Description
Boolean	Backlight compensation state.

VIDYOLOCALCAMERA : GETFRAMERATETRADEOFFPROFILE

Gets the frame rate v.s. video quality tradeoff profile.

Syntax

```
GetFramerateTradeOffProfile()
```

Returns

Type	Description
	The frame rate tradeoff profile.

VIDYOLOCALCAMERA : GETID

Gets the unique ID of the camera.

Syntax

```
GetId()
```

Returns

Type	Description
String	Unique ID of the camera.

VIDYOLOCALCAMERA : GETNAME

Gets the name of the camera.

Syntax

```
GetName()
```

Returns

Type	Description
String	Name of the camera.

VIDYOLOCALCAMERA : GETPOSITION

Gets the position of the camera.

Syntax

```
GetPosition()
```

Returns

Type	Description
	Position of the camera.

VIDYOLOCALCAMERA : GETPREVIEWLABEL

Gets the preview label of the camera.

Syntax

```
GetPreviewLabel()
```

Returns

Type	Description
String	Display name of the camera.

VIDYOLOCALCAMERA : GETRESOLUTIONTRADEOFFPROFILE

Gets the resolution v.s. video quality tradeoff profile.

Syntax

```
GetResolutionTradeOffProfile()
```

Returns

Type	Description
	The resolution tradeoff profile.

VIDYOLOCALCAMERA : ISCONTROLDIGITAL

Checks if the camera control to digital only.

Syntax

```
IsControlDigital()
```

Returns

Type	Description
Boolean	LMI_TRUE is digital only mode is set, LMI_FALSE otherwise.

VIDYOLOCALCAMERA : ISSUSPENDED

Checks if the camera is suspended.

Syntax

```
IsSuspended()
```

Returns

Type	Description
Boolean	LMI_TRUE is suspended, LMI_FALSE otherwise.

VIDYOLOCALCAMERA : SETASPECTRATIOCONSTRAINT

Sets the aspect ratio constraint on the camera which will adjust local camera frames to match.

Syntax

```
SetAspectRatioConstraint({aspectRatioWidth:Number, aspectRatioHeight:Number})
```

Parameters

Name	Type	Description
aspectRatioWidth	Number	Width of the aspect ratio.

aspectRatioHeight	Number	Height of the aspect ration.
-------------------	--------	------------------------------

Returns

Type	Description
Boolean	LMI_TRUE on success.

VIDYOLOCALCAMERA : SETBACKLIGHTCOMPENSATION

Set the backlight compensation which improves the dynamic range of the image when the subject is dark but the background is bright.

Syntax

```
SetBacklightCompensation({backlightCompensation: Boolean})
```

Parameters

Name	Type	Description
backlightCompensation	Boolean	Compensate for the bright backlighting.

Returns

Type	Description
Boolean	LMI_TRUE on success.

VIDYOLOCALCAMERA : SETCONTROLDIGITAL

Sets the camera control to digital only.

Syntax

```
SetControlDigital({digital: Boolean})
```

Parameters

Name	Type	Description
digital	Boolean	Forces digital control.

Returns

Type	Description
Void	

VIDYOLOCALCAMERA : SETFRAMERATETRADEOFFPROFILE

Sets the frame rate v.s. video quality tradeoff profile.

Syntax

```
SetFramerateTradeOffProfile({profile:String})
```

Parameters

Name	Type	Description
profile	String	Frame-rate tradeoff profile.

Returns

Type	Description
Boolean	LMI_TRUE on success.

VIDYOLOCALCAMERA : SETMAXCONSTRAINT

Sets the maximum resolution and framerate constraint on the local camera. This overrides the defaults which are based on the CPU profile of the endpoint. The maximum if not an exact match for the camera resolution will be chosen by the approximate closest resolution based on calculations that include the width, height and frameInterval.

Syntax

```
SetMaxConstraint({width:Number, height:Number, frameInterval:Number})
```

Parameters

Name	Type	Description
width	Number	Maximum width.
height	Number	Maximum height.
frameInterval	Number	Maximum frame interval. Setting to 0 selects the fastest frame rate.

Returns

Type	Description
Boolean	LMI_TRUE on success.

VIDYOLOCALCAMERA : SETNUDGETIMES

Set the time delta that the camera moves left/right (pan), up/down (tilt), and/or in/out (zoom) for each nudge if the camera supports only start/stop (relative) behavior.

Syntax

```
SetNudgeTimes({panTime:Number, tiltTime:Number, zoomTime:Number})
```

Parameters

Name	Type	Description
panTime	Number	The time delta to pan the camera per nudge.
tiltTime	Number	The time delta to tilt the camera per nudge.
zoomTime	Number	The time delta to zoom the camera per nudge.

Returns

Type	Description
Boolean	LMI_TRUE when applied, LMI_FALSE otherwise.

VIDYOLOCALCAMERA : SETPREVIEWLABEL

Set the preview label of the camera.

Syntax

```
SetPreviewLabel({previewLabel:String})
```

Parameters

Name	Type	Description
previewLabel	String	Display name to set on the camera.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE otherwise.

VIDYOLOCALCAMERA : SETRESOLUTIONTRADEOFFPROFILE

Sets the resolution v.s. video quality trade off profile.

Syntax

```
SetResolutionTradeOffProfile({profile:String})
```

Parameters

Name	Type	Description
profile	String	Resolution tradeoff profile.

Returns

Type	Description
Boolean	LMI_TRUE on success.

VIDYOLOCALCAMERA : SETTARGETBITRATE

Sets the target send bit rate for a local camera.

Syntax

```
SetTargetBitRate({bitRate: Number})
```

Parameters

Name	Type	Description
bitRate	Number	The target send bit rate in bits per second.

Returns

Type	Description
Void	

VIDYOLOCALCAMERA : SHOWCAMERACONTROL

Shows camera control when rendering preview.

Syntax

```
ShowCameraControl({show: Boolean})
```

Parameters

Name	Type	Description
show	Boolean	Show or hide camera controls.

Returns

Type	Description
Boolean	LMI_TRUE on success.

LOCALMICROPHONE

This object represents a microphone on the local endpoint.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.

METHODS

VIDYOLOCALMICROPHONE : GETID

Gets the unique ID of the microphone.

Syntax

```
GetId()
```

Returns

Type	Description
String	Unique ID of the microphone.

VIDYOLOCALMICROPHONE : GETNAME

Gets the name of the microphone.

Syntax

```
GetName()
```

Returns

Type	Description
String	Name of the microphone.

VIDYOLOCALMICROPHONE : GETSIGNALTYPE

Gets signal type of the microphone, which affects how the audio is processed.

Syntax

```
GetSignalType()
```

Returns

Type	Description
	The signal type of the microphone.

VIDYOLOCALMICROPHONE : SETSIGNALTYPE

Sets the signal type of the microphone which changes how the audio is processed. If this API will be called for selected microphone this will cause a force reconfiguration of the device and will restart the audio capturer.

Syntax

```
SetSignalType({signalType:String})
```

Parameters

Name	Type	Description
signalType	String	The signal type of the microphone.

Returns

Type	Description
Boolean	LMI_TRUE on success. Also after the configuration will be applied onLocalMicrophoneStateUpdated will be called with VIDYO_DEVICESTATE_ConfigureSuccess on success, VIDYO_DEVICESTATE_ConfigureError otherwise.

LOCALMONITOR

This object represents a monitor on the local endpoint.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.

METHODS

VIDYOLOCALMONITOR : GETID

Gets the unique ID of the monitor.

Syntax

```
GetId()
```

Returns

Type	Description
String	Unique ID of the monitor.

VIDYOLOCALMONITOR : GETNAME

Gets the name of the monitor.

Syntax

```
GetName()
```

Returns

Type	Description
String	Name of the monitor.

VIDYOLOCALMONITOR : SETFRAMEINTERVAL

Syntax

```
SetFrameInterval({frameInterval: Number})
```

Parameters

Name	Type	Description
frameInterval	Number	The interval in nanoseconds between consecutive frames. Default is 333333333 which is 3 frames per second.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE otherwise.

VIDYOLOCALMONITOR : SETLOWLATENCYPROFILEBETA

Sets the low latency profile of the monitor which treats the stream as a regular video source. This prioritizes latency over quality.

Syntax

```
SetLowLatencyProfile({profile: Boolean})
```

Parameters

Name	Type	Description
profile	Boolean	Low latency profile toggle. Default is LMI_FALSE.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE otherwise.

LOCALSPEAKER

This object represents a speaker on the local endpoint.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.

METHODS

VIDYOLOCALSPEAKER : GETID

Gets the unique ID of the speaker.

Syntax

```
GetId()
```

Returns

Type	Description
String	Unique ID of the speaker.

VIDYOLOCALSPEAKER : GETNAME

Gets the name of the speaker.

Syntax

```
GetName()
```

Returns

Type	Description
String	Name of the speaker.

LOCALWINDOWSHARE

This object represents a window share on the local endpoint.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
applicationName	String	Name of the application that owns the window.
id	String	Unique ID.

name	String	Name of the window share.
------	--------	---------------------------

METHODS

VIDYOLOCALWINDOWSHARE : GETAPPLICATIONNAME

Gets the name of the application that owns the window.

Syntax

```
GetApplicationName()
```

Returns

Type	Description
String	The name of the application that owns the window

VIDYOLOCALWINDOWSHARE : GETID

Gets the unique ID of the window share.

Syntax

```
GetId()
```

Returns

Type	Description
String	Unique ID of the window share.

VIDYOLOCALWINDOWSHARE : GETNAME

Gets the name of the window share.

Syntax

```
GetName()
```

Returns

Type	Description
String	Name of the window share.

VIDYOLOCALWINDOWSHARE : GETPREVIEWLABEL

Gets the preview label of the window share.

Syntax

```
GetPreviewLabel()
```

Returns

Type	Description
String	Display name of the window share.

VIDYOLOCALWINDOWSHARE : GETPROCESSID

Gets the ID of the process that owns the window share.

Syntax

```
GetProcessId()
```

Returns

Type	Description
Number	ID of the process that owns the window share.

VIDYOLOCALWINDOWSHARE : GETPROCESSNAME

Syntax

```
GetProcessName()
```

Returns

Type	Description
String	

VIDYOLOCALWINDOWSHARE : GETUPDATEDNAME

Retrieved the up to date name of the Window which might have changed since it was created.

Syntax

```
GetUpdatedName()
```

Returns

Type	Description
String	

VIDYOLOCALWINDOWSHARE : ISAPPLICATIONNAMESET

Checks if the name of the application name is set.

Syntax

```
IsApplicationNameSet()
```

Returns

Type	Description
Boolean	LMI_TRUE when the applicaiton name is set, LMI_FALSE otherwise.

VIDYOLOCALWINDOWSHARE : ISNAMESET

Checks if the name of the window share is set.

Syntax

```
IsNameSet()
```

Returns

Type	Description
Boolean	LMI_TRUE when the window share name is set, LMI_FALSE otherwise.

VIDYOLOCALWINDOWSHARE : SETFRAMEINTERVAL

Sets the maximum frame interval that will be used to scrape the window share

Syntax

```
SetFrameInterval({frameInterval: Number})
```

Parameters

Name	Type	Description
frameInterval	Number	The interval in nanoseconds between consecutive frames. Default is 333333333 which is 3 frames per second.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE otherwise.

VIDYOLOCALWINDOWSHARE : SETLOWLATENCYPROFILEBETA

Sets the low latency profile of the window share which treats the stream as a regular video source. This prioritizes latency over quality.

Syntax

```
SetLowLatencyProfile({profile: Boolean})
```

Parameters

Name	Type	Description
profile	Boolean	Low latency profile toggle. Default is LMI_FALSE.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE otherwise.

VIDYOLOCALWINDOWSHARE : SETPREVIEWLABEL

Set the preview label of the window share.

Syntax

```
SetPreviewLabel({previewLabel: String})
```

Parameters

Name	Type	Description
previewLabel	String	Display name to set on the window share.

Returns

Type	Description
Boolean	LMI_TRUE on success, LMI_FALSE otherwise.

REMOTECAMERA

This object represents a camera which belongs to the remote participant.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.

METHODS

VIDYOREMOTECAMERA : CONTROLPTZ

Send the PTZ Nudge commands to the remote camera.

Syntax

```
ControlPTZ({pan:Number, tilt:Number, zoom:Number})
```

Parameters

Name	Type	Description
pan	Number	Number of nudges to pan the camera.

tilt	Number	Number of nudges to tilt the camera.
zoom	Number	Number of nudges to zoom the camera.

Returns

Type	Description
Boolean	LMI_TRUE when the control command was sent, LMI_FALSE otherwise.

VIDYOREMOTECAMERA : GETID

Gets the unique ID of the camera.

Syntax

```
GetId()
```

Returns

Type	Description
String	Unique ID of the camera.

VIDYOREMOTECAMERA : GETNAME

Gets the name of the camera.

Syntax

```
GetName()
```

RETURNS

Type	Description
String	Name of the camera.

VIDYOREMOTECAMERA : GETPOSITION

Gets the position of the camera.

Syntax

```
GetPosition()
```

Returns

Type	Description
	Position of the camera.

VIDYOREMOTECAMERA : ISCONTROLLABLE

Checks if the remote camera is controllable.

Syntax

```
IsControllable()
```

Returns

Type	Description
Boolean	LMI_TRUE if controllable, LMI_FALSE otherwise.

VIDYOREMOTECAMERA : SHOWCAMERACONTROL

Shows camera control when rendering remote camera.

Syntax

```
ShowCameraControl({show: Boolean})
```

Parameters

Name	Type	Description
show	Boolean	Show or hide camera controls.

Returns

Type	Description
Boolean	LMI_TRUE on success.

REMOTE MICROPHONE

This object represents a microphone which belongs to the remote participant.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.
signalType	String	Signal Type.

METHODS

VIDYOREMOTEMICROPHONE : GETID

Gets the unique ID of the microphone.

SYNTAX

```
GetId()
```

Returns

Type	Description
String	Unique ID of the microphone.

VIDYOREMOTEMICROPHONE : GETNAME

Gets the name of the microphone.

Syntax

```
GetName()
```

RETURNS

Type	Description
String	Name of the microphone.

VIDYOREMOTEMICROPHONE : GETSIGNALTYPE

Gets signal type of the microphone, which affects how the audio is processed.

Syntax

```
GetSignalType()
```

Returns

Type	Description
	The signal type of the microphone.

REMOTEWINDOWSHARE

This object represents a window share which belongs to the remote participant.

DESTRUCTOR

Syntax

Destruct()

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	Unique ID.
name	String	Name.

METHODS

VIDYOREMOTEWINDOWSHARE : GETID

Gets the unique ID of the window share.

Syntax

GetId()

Returns

Type	Description
String	Unique ID of the window share.

VIDYOREMOTEWINDOWSHARE : GETNAME

Gets the name of the window share.

Syntax

GetName()

Returns

Type	Description
String	Name of the window share.

VIDEOFRAME

This object represents a video frame.

CONSTRUCTOR

Syntax

```
CreateVidyoVideoFrame({format:String, buffer:Number, size:Number,
releaseCallback:function, width:Number, height:Number})
```

Parameters

Name	Type	Description
format	String	The media format of the video frame.
buffer	Number	A pointer to externally allocated memory that holds the pixel data.
size	Number	The size, in bytes, of the frame data.
releaseCallback	function	A callback to be called on when external buffer may be reclaimed.
width	Number	The width of the video media frame in pixels.
height	Number	The height of the video media frame in pixels.

Returns

Type	Description
------	-------------

Object	Returns a pointer to a constructed object on success, or NULL on failure.
---------------	---

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
data	Number	Pointer to the frame's payload.
dataCb	Number	Pointer to the frame's payload.
dataCr	Number	Pointer to the frame's payload.
dataY	Number	Pointer to the frame's payload.
elapsedTime	Number	Elapsed time since the beginning of frame capture that is increased by the duration of each frame.
format	String	Format of the video frame.
height	Number	Height of the video frame.
heightCb	Number	Height in bytes of the frame's Cb pixel data.
heightCr	Number	Height in bytes of the frame's Cr pixel data.
heightY	Number	Height in bytes of the frame's Y pixel data.

offsetCb	Number	Number of bytes from the plane's payload to the start of the pixel data.
offsetCr	Number	Number of bytes from the plane's payload to the start of the pixel data.
offsetY	Number	Number of bytes from the plane's payload to the start of the pixel data.
pitchCb	Number	Number of bytes between consecutive pixels in a row of the plane.
pitchCr	Number	Number of bytes between consecutive pixels in a row of the plane.
pitchY	Number	Number of bytes between consecutive pixels in a row of the plane.
size	Number	Size in bytes of the frame's payload.
sizeCb	Number	Size in bytes of the plane's payload.
sizeCr	Number	Size in bytes of the plane's payload.
sizeY	Number	Size in bytes of the plane's payload.
timestamp	Number	Wallclock timestamp associated with the creation of the audio frame object.
width	Number	Width of the video frame.
widthCb	Number	Width in bytes of the frame's Cb pixel data.
widthCr	Number	Width in bytes of the frame's Cr pixel data.
widthY	Number	Width in bytes of the frame's Y pixel data.

METHODS

VIDYOVIDEOFRAME : GETDATACB

Syntax

```
GetDataCb()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETDATAcr

Syntax

```
GetDataCr()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETDATAy

Syntax

```
GetDataY()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETELAPSEDTIME

Syntax

```
GetElapsedTime()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETFORMAT

Gets the format of the frame.

Syntax

```
GetFormat()
```

Returns

Type	Description
String	Format of the video frame.

VIDYOVIDEOFRAME : GETHEIGHT

Gets the height of the frame.

Syntax

```
GetHeight()
```

Returns

Type	Description
Number	Height of the frame.

VIDYOVIDEOFRAME : GETHEIGHTCB

Syntax

```
GetHeightCb()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETHEIGHTCR

Syntax

```
GetHeightCr()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETHEIGHTY

Syntax

```
GetHeightY()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETOFFSETCB

Syntax

```
GetOffsetCb()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETOFFSETCR

Syntax

```
GetOffsetCr()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETOFFSETY

Syntax

```
GetOffsetY()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETPITCHCB

Syntax

```
GetPitchCb()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETPITCHCR

Syntax

```
GetPitchCr()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETPITCHY

Syntax

```
GetPitchY()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETSIZE

Syntax

```
GetSize()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETSIZECb

Syntax

```
GetSizeCb()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETSIZECr

Syntax

```
GetSizeCr()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETSIZEY

Syntax

```
GetSizeY()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETTIMESTAMP

Syntax

```
GetTimestamp()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETWIDTH

Gets the width of the frame.

Syntax

```
GetWidth()
```

Returns

Type	Description
Number	Width of the frame.

VIDYOVIDEOFRAME : GETWIDTHCb

Syntax

```
GetWidthCb()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETWIDTHCR

Syntax

```
GetWidthCr()
```

Returns

Type	Description
Number	

VIDYOVIDEOFRAME : GETWIDTHY

Syntax

```
GetWidthY()
```

Returns

Type	Description
Number	

CHATMESSAGE

Contains the text message and additional information about who sent it and when.

PROPERTIES

Name	Type	Description
body	String	The body of the VidyoChatMessage.

id	Number	Unique ID of the VidyoChatMessage.
senderType	String	The type of the sender.
timestamp	Number	The time when the message was created.
type	String	The type of VidyoChatMessage.
userId	String	Unique ID of a VidyoParticipant who sent the message or NULL depending on the VidyoChatMessageSenderType.

LOGRECORD

A record of one logged event.

PROPERTIES

Name	Type	Description
categoryName	Number	undefined
eventTime	Number	undefined
file	String	undefined
functionName	String	undefined
level	String	undefined
line	Number	undefined
message	String	undefined
name	String	undefined
threadName	String	undefined

PARTICIPANT

This object represents a participant in the room or call.

DESTRUCTOR

Syntax

```
Destruct()
```

Returns

Type	Description
Void	

PROPERTIES

Name	Type	Description
id	String	ID that is unique to the room or call only.
name	String	Display name of the participant.
trust	String	undefined
userId	String	ID that is globally unique such as user@domain.

METHODS

VIDYOPARTICIPANT : GETAPPLICATIONTYPE

Syntax

```
GetApplicationType()
```

Returns

Type	Description
	one of the enum value from VidyoparticipantApplicationType

VIDYOPARTICIPANT : GETID

Gets the ID that is unique to the room or call only.

Syntax

```
GetId()
```

Returns

Type	Description
String	ID that is unique to the room or call only.

VIDYOPARTICIPANT : GETNAME

Gets the display name of the participant.

Syntax

```
GetName()
```

Returns

Type	Description
String	Display name of the participant.

VIDYOPARTICIPANT : GETTRUST

Syntax

```
GetTrust()
```

Returns

Type	Description

VIDYOPARTICIPANT : GETUSERID

Gets the ID that is globally unique such as user@domain.

Syntax

```
GetUserId()
```

Returns

Type	Description
String	ID that is globally unique such as user@domain.

VIDYOPARTICIPANT : ISHIDDEN

Gets the hidden state of the participant.

Syntax

```
IsHidden()
```

Returns

Type	Description
Boolean	LMI_TRUE if the participant is hidden, LMI_FALSE otherwise.

VIDYOPARTICIPANT : ISRECORDING

Gets the recording state of the participant.

Syntax

```
IsRecording()
```

Returns

Type	Description
------	-------------

Boolean	LMI_TRUE if the participant is recording, LMI_FALSE otherwise.
---------	--

VIDYOPARTICIPANT : ISSELECTABLE

Gets the selectable state of the participant.

Syntax

```
IsSelectable()
```

Returns

Type	Description
Boolean	LMI_TRUE if the participant is selectable, LMI_FALSE otherwise.

BANDWIDTHSUMMARYSTATS

Statistics for an endpoint rate shaper.

PROPERTIES

Name	Type	Description
actualEncoderBitRate	Number	Actual encoder bitrate.
availableBandwidth	Number	Available bandwidth.
leakyBucketDelay	Number	Leaky bucket delay.
retransmitBitRate	Number	Bitrate used for retransmission.
targetEncoderBitRate	Number	Target encoder bitrate.
totalTransmitBitRate	Number	Total transmit bitrate.

ENDPOINTSTATS

Statistics for an endpoint.

PROPERTIES

Name	Type	Description
applicationTag	String	Tag set by the application.
buildTag	String	Tag from the build system.
bytesReceivedTcp	Number	Total number of bytes received over TCP.
bytesReceivedUdp	Number	Total number of bytes received over UDP.
bytesSentTcp	Number	Total number of bytes sent over TCP.
bytesSentUdp	Number	Total number of bytes sent over UDP.
connectTime	String	Gregorian Time of the last connect attempt.
id	String	Room ID.
libraryVersion	String	Version of the library.
localCameraStats	Array	Vector of stats of cameras.
localMicrophoneStats	Array	Vector of stats of microphones.
localMonitorStats	Array	Vector of stats of monitors.
localRendererStats	Array	Vector of stats of renderers.
localSpeakerStats	Array	Vector of stats of speakers.
localWindowShareStats	Array	Vector of stats of window shares.
logStats	Object	Stats of the log.
loginTimeConsumedMs	Number	Number of milliseconds consumed in user login process.
maxBitRate	Number	Max send bit rate for all srouces.

maxEncodePixelRateInitial	Number	Max encode pixel rate when resource manager uses the default profile.
mediaEnableTimeConsumedMs	Number	Number of milliseconds consumed in media enable process.
mediaRouteAcquireTimeConsumedMs	Number	Number of milliseconds consumed in media route acquire process.
networkInterfaceStats	Array	Network interfaces on the endpoint.
osName	String	Name of the operating system.
osVersion	String	Version of the operating system.
processorBrand	String	CPU brand.
processorLogicalCores	Number	Number of logical cores in the CPU (including hyperthreading).
processorModel	String	CPU Model.
processorPhysicalCores	Number	Number of physical cores in the CPU (ignoring hyperthreading).
processorSpeed	Number	CPU frequency, in MHz.
roomEnterTimeConsumedMs	Number	Number of milliseconds consumed in room entering process.
timeStamp	String	Gregorian Timestamp when the stats were measured.
userStats	Array	Vector of stats of users.
virtualVideoSourceStats	Array	Vector of stats of virtual window shares.

LATENCYTESTDATASTATS

Statistics for a latency test.

PROPERTIES

Name	Type	Description
address	String	Address of the latency test host.
latencyMs	Number	Latency to the host in milliseconds.
latitude	Number	Latitude of the host.
longitude	Number	Longitude of the host.
name	String	Name of the latency test host.
responseReceived	Boolean	Response has been received from the host.

LATENCYTESTSTATS

Statistics for a latency test.

PROPERTIES

Name	Type	Description
latencyTestDataStats	Array	Vector of stats data of latency test.

LOCALMICROPHONESTATS

Statistics for a local microphone.

PROPERTIES

Name	Type	Description
aecEchoCoupling	Number	AEC echo coupling.
agcAverageGain	Number	AGC average gain.
bitsPerSample	Number	Bits per sample.
format	String	Format.

id	String	Id.
localSpeakerStreams	Array	Vector of stats of local speaker streams.
name	String	Name.
noiseSuppressionSnr	Number	Noise suppression SNR.
numberOfChannels	Number	Number of channels.
remoteSpeakerStreams	Array	Vector of stats of remote speaker streams.
sampleRateMeasured	Number	Measured sample rate.
sampleRateSet	Number	Sample rate setting.

LOCALRENDERERSTATS

Statistics for a local renderer.

PROPERTIES

Name	Type	Description
format	String	Format.
frameIntervalMeasured	Number	Frame interval measured by the renderer.
frameIntervalSet	Number	Frame interval setting.
height	Number	Height.
id	String	Id.
name	String	Name.
totalFrames	Number	Total frames rendered.
totalPixels	Number	Total pixels rendered.

width	Number	Width.
-------	--------	--------

LOCALRENDERERSTREAMSTATS

Statistics for a local renderer.

PROPERTIES

Name	Type	Description
bufferDropped	Number	Dropped frames from video buffer.
bufferSize	Number	Size of video buffer.
height	Number	Height.
id	String	Id.
name	String	Name.
width	Number	Width.

LOCALSPEAKERSTATS

Statistics for a local speaker.

PROPERTIES

Name	Type	Description
bitsPerSample	Number	Bits per sample.
format	String	Audio format.
id	String	Id.
name	String	Name.
numberOfChannels	Number	Number of channels.

sampleRateMeasured	Number	Sample rate measured.
sampleRateSet	Number	Sample rate setting.

LOCALSPEAKERSTREAMSTATS

Statistics for a local speaker.

PROPERTIES

Name	Type	Description
bitsPerSample	Number	Bits per sample.
delay	Number	Audio delay in the stream jitter buffer.
format	String	Audio format.
highThreshold	Number	High threshold of stream jitter buffer over which the stream will be gently pared down to get into the proper range.
highestThreshold	Number	Highest threshold of stream jitter buffer over which the stream will be aggressively pared down to get into the proper range.
isActive	Boolean	Active stream status.
lastEnergy	Number	Last energy in dBFS of the stream.
lastPlayed	Number	Last time audio was played from the stream.
lowThreshold	Number	Low threshold of stream jitter buffer under which the stream will be gently padded to get into the proper range.
lowestThreshold	Number	Lowest threshold of stream jitter buffer under which the stream will be aggressively padded to get into the proper range.
maxThreshold	Number	Maximum threshold of stream jitter buffer over which the stream will be aggressively pared down by dropping non-speech frames to get into the proper range.
name	String	Name.

numberOfChannels	Number	Number of channels.
overrun	Number	Amount of time stream was over the maxThreshold.
played	Number	Amount of audio played by the stream.
sampleRate	Number	Sample rate.
underrun	Number	Amount of time stream was in the empty but active state.

LOCALVIDEOSOURCESTATS

Statistics for a local video source.

PROPERTIES

Name	Type	Description
format	String	Format.
frameIntervalMeasured	Number	Frame interval measured between last 2 frames.
frameIntervalSet	Number	Frame interval setting.
height	Number	Height.
id	String	Id.
localRendererStreams	Array	Vector of stats of local renderer streams.
name	String	Name.
remoteRendererStreams	Array	Vector of stats of remote renderer streams.
targetBitRate	Number	Target bit rate.
totalFrames	Number	Total frames.
width	Number	Width.

LOGDATASTATS

Statistics for a log function:line number.

PROPERTIES

Name	Type	Description
name	String	Function name and line number of the log record.
occurrences	Number	Number of occurrences of this log record.

LOGSTATS

Statistics of the endpoint logs.

PROPERTIES

Name	Type	Description
logErrorDataStats	Array	Vector of error stats data of logs.
logWarningDataStats	Array	Vector of warning stats data of logs.

MEDIACONNECTIONTRANSPORTINFO

Information about media connection transport.

PROPERTIES

Name	Type	Description
addressType	String	Address type.
componentType	String	Transport component type.
connectionId	Number	Connection ID.
connectionType	String	Transport connection type.

externalLocalAddr	String	External local address.
externalRemoteAddr	String	External remote address.
interfaceName	String	Interface name.
interfaceType	String	Interface type.
internalLocalAddr	String	Internal local address.
internalRemoteAddr	String	Internal remote address.
transportPlugIn	String	Transport plugin.

NETWORKINTERFACESTATS

Statistics for a network interface.

PROPERTIES

Name	Type	Description
isUp	Boolean	State of the interface.
name	String	Interface name.
type	String	Interface type such as wifi or wired.

PARTICIPANTGENERATIONSTATS

Statistics for a participant generation, ordered list of dynamic participants.

PROPERTIES

Name	Type	Description
cameraId	String	Camera id.
cameraName	String	Camera name.

enabled	Boolean	State of the generation.
frameInterval	Number	Frame interval setting.
height	Number	Height.
id	String	Participant id.
name	String	Participant name.
pixelRate	Number	Pixel rate to ask for.
width	Number	Width.

PARTICIPANTSTATS

Statistics for a room.

PROPERTIES

Name	Type	Description
id	String	Participant ID.
name	String	Current CPU usage.
remoteCameraStats	Array	Vector of stats of cameras.
remoteMicrophoneStats	Array	Vector of stats of microphones.
remoteWindowShareStats	Array	Vector of stats of speakers.
userId	String	ID associated with the VidyoUser of that participant.

RATESHAPERSTATS

Statistics for an endpoint rate shaper.

PROPERTIES

Name	Type	Description
delayNormal	Number	Normal delay.
delayRetransmit	Number	Retransmit delay.
dropNormal	Number	Dropped normal.
packetsNormal	Number	Normal packets.
packetsRetransmit	Number	Retransmitted packets.

REMOTE MICROPHONE STATS

Statistics for a remote microphone.

PROPERTIES

Name	Type	Description
bitsPerSample	Number	Bits per sample.
codecDtx	Number	Codec DTX.
codecName	String	Codec name.
codecQualitySetting	Number	Codec quality setting.
id	String	Id.
lastFrameMs	Number	Milliseconds since last received frame.
localSpeakerStreams	Array	Vector of stats of local speaker streams.
name	String	Name.
numberOfChannels	Number	Number of channels.
receiveNetworkBitRate	Number	Bandwidth of receive network.

receiveNetworkDelay	Number	Delay in receive network.
receiveNetworkDroppedPackets	Number	Dropped packets in receive network.
receiveNetworkJitter	Number	Jitter in receive network.
receiveNetworkPacketsConcealed	Number	Concealed packets in receive network.
receiveNetworkPacketsLost	Number	Lost packets in receive network.
sampleRateMeasured	Number	Measured sample rate.
sampleRateSet	Number	Sample rate setting.

REMOTE RENDERER STREAM STATS

Statistics for a remote renderer.

PROPERTIES

Name	Type	Description
codecFir	Number	Codec FIR.
codecIframes	Number	Codec I frames.
codecLayers	Number	Codec layers.
codecNacks	Number	Codec nacks.
codecName	String	Codec name.
fps	Number	Frame rate determined by the operating point.
fpsInput	Number	Frame rate at encoder input.
fpsSent	Number	Frame rate at the encoder output.
framesDropped	Number	Number of frames dropped due to overconstraint or resizing errors.

height	Number	Height determined by the operating point.
lastHeight	Number	Height of the last frame received from the capturing device.
lastWidth	Number	Width of the last frame received from the capturing device.
name	String	Name.
sendNetworkBitRate	Number	Send bitrate.
sendNetworkRtt	Number	Round trip time of send network.
width	Number	Width determined by the operating point.

REMOTE SPEAKER STREAM STATS

Statistics for a remote speaker.

PROPERTIES

Name	Type	Description
bitsPerSample	Number	Bits per sample.
codecDtx	Number	Codec DTX.
codecName	String	Codec name.
codecQualitySetting	Number	Codec quality setting.
name	String	Name.
numberOfChannels	Number	Number of channels.
sampleRate	Number	Sample rate.
sendNetworkBitRate	Number	Send bitrate.
sendNetworkRtt	Number	Round trip time of send network.

REMOTEVIDEOSOURCESTATS

Statistics for a remote video source.

PROPERTIES

Name	Type	Description
codecFir	Number	Codec FIR.
codecIframes	Number	Codec I frames.
codecLayers	Number	Codec layers.
codecNacks	Number	Codec nacks.
codecName	String	Codec name.
fpsDecoded	Number	Frame rate of decoder output.
fpsDecoderInput	Number	Frame rate at decoder input.
fpsRendered	Number	Frame rate of renderer.
height	Number	Height.
id	String	Id.
lastFrameMs	Number	Milliseconds since last received frame.
localRendererStreams	Array	Vector of stats of local renderer streams.
maxStreamFrameRate	Number	Width.
maxStreamHeight	Number	Width.
maxStreamWidth	Number	Width.
name	String	Name.

receiveNetworkBitRate	Number	Source bitrate.
receiveNetworkPacketsConcealed	Number	Number of packets concealed.
receiveNetworkPacketsLost	Number	Number of lost packets.
receiveNetworkPacketsReordered	Number	Number of packets reordered.
receiveNetworkRecoveredWithFec	Number	Number of packets recovered with FEC.
showFrameRate	Number	Width.
showHeight	Number	Width.
showPixelRate	Number	Width.
showState	String	State of how the source is received from the server (on/off/auto).
showWidth	Number	Width.
videoDeltaFromAudioMs	Number	Last known delta between audio and video before de-jittering in milliseconds. Negative number means audio is behind video.
width	Number	Width.

ROOMSTATS

Statistics for a room.

PROPERTIES

Name	Type	Description
availableDecodeBwPercent	Number	Available Percent of the Bandwidth for the decoder.
availableDecodeCpuPercent	Number	Available Percent of the CPU for the decoder.
availableEncodeBwPercent	Number	Available Percent of the Bandwidth for the encoder.
availableEncodeCpuPercent	Number	Available Percent of the CPU for the encoder.

bandwidthApp	Object	Bandwidth summary statistics of application content.
bandwidthAudio	Object	Bandwidth summary statistics of audio content.
bandwidthVideo	Object	Bandwidth summary statistics of video content.
callId	String	Unique ID every time media is started and stopped.
conferenceId	String	ID conference on the reflector associated with this VidyoRoom.
cpuUsage	Number	Current CPU usage.
currentBandwidthDecodePixelRate	Number	Current Bandwidth decode pixel rate.
currentBandwidthEncodePixelRate	Number	Current Bandwidth encode pixel rate.
currentCpuDecodePixelRate	Number	Current CPU decode pixel rate.
currentCpuEncodePixelRate	Number	Current CPU encode pixel rate.
id	String	Room ID.
maxDecodePixelRate	Number	Max decode pixel rate.
maxEncodePixelRate	Number	Max encode pixel rate.
maxVideoSources	Number	Maximum video sources allowed based on the minimum of configured and constrained by resource manager.
participantGenerationStats	Array	Ordered list of generations (selected participants).
participantStats	Array	Vector of stats of participants.
rateShaperApp	Object	Rate shaper statistics of application content.
rateShaperAudio	Object	Rate shaper statistics of audio content.
rateShaperVideo	Object	Rate shaper statistics of video content.

receiveBitRateAvailable	Number	Estimated available receive bandwidth in bps as detected by the automatic bandwidth detection algorithm.
receiveBitRateTotal	Number	Estimate of the current receive bit rate on the entire media session.
reflectorId	String	ID of the reflector to which the media connection is established.
sendBitRateAvailable	Number	Estimated available send bandwidth in bps as detected by the automatic bandwidth detection algorithm.
sendBitRateTotal	Number	Estimate of the current send bit rate for the entire media session.
staticSources	Number	Current amount of static sources shown.
transportInformation	Array	Vector of media transport information.

USERSTATS

Statistics for an user.

PROPERTIES

Name	Type	Description
host	String	Login host.
id	String	User ID.
latencyTestStats	Object	Stats of latency test.
port	Number	Login Port.
roomStats	Array	Vector of stats of rooms.
serviceType	String	Login service type.