

# VaxVoIP

SIP SOFTPHONE SDK

**SIP SOFTPHONE SDK**  
Microsoft Windows Desktop OS  
TECHNICAL DOCUMENTATION  
VERSION 2.4

**CONTENTS**

**INTRODUCTION AND QUICK START ..... 6**

**EXPORTED FUNCTIONS ..... 7**

InitializeEx() ..... 7

RegisterToProxy() ..... 10

UnRegisterToProxy()..... 11

OpenLine() ..... 12

CloseLine()..... 13

IsLineBusy() ..... 15

IsLineConnected() ..... 16

SetLicenceKey() ..... 17

GetVaxObjectError() ..... 18

ConnectCall()..... 20

Disconnect() ..... 22

DialCall()..... 23

GetAudioInDevTotal() ..... 24

GetAudioOutDevTotal() ..... 25

GetAudioInDevName() ..... 26

GetAudioOutDevName()..... 27

AcceptCall()..... 28

RejectCall() ..... 29

TransferCallEx() ..... 30

JoinTwoLine() ..... 31

HoldLine() ..... 32

IsLineHold() ..... 33

UnHoldLine() ..... 34

EnableKeepAlive() ..... 35

DisableKeepAlive() ..... 36

DeselectAllVoiceCodec() ..... 37

SelectAllVoiceCodec() ..... 38

GetOutboundCodec() ..... 39

GetInboundCodec() ..... 40

SelectVoiceCodec()..... 41

DeselectVoiceCodec() ..... 42

GetMyIP() ..... 43

GetStartMyIP() ..... 44

GetNextMyIP()..... 45

DigitDTMF()..... 46

SetDTMFVolume() ..... 47

GetDTMFVolume() ..... 48

EnableForceInbandDTMF() ..... 49

DisableForceInbandDTMF()..... 50

DetectAMD()..... 51

EnableEchoNoiseCancellation() ..... 52

DisableEchoNoiseCancellation()..... 53

EnableAGC() ..... 54

DisableAGC() ..... 55

IsRecording() ..... 56

StartRecording()	57
StopRecording()	58
ResetRecording()	59
SaveRecordingToWaveFile()	60
IsWaveFilePlaying()	61
PlayWaveOpen()	62
PlayWaveClose()	63
PlayWaveStart()	64
PlayWaveSkipTo	65
PlayWavePosition()	66
PlayWaveTotalTime()	67
PlayWavePause()	68
PlayWaveStop ()	69
MuteLineSPK()	70
MuteLineMIC()	71
MuteSpk()	72
MuteMic()	73
GetSpkVolume( )	74
SetSpkVolume()	75
MicVolume()	76
SetMicVolume()	77
EnableMicBoost()	78
DisableMicBoost()	79
IsMicBoostEnable()	80
EnableDonotDisturb()	81
DisableDonotDisturb()	82
GetMicSoundLevel()	83
GetSpkSoundLevel()	84
SetSessionLostTick()	85
SetSpkSoftVolume()	86
SetUserAgentSIP()	87
GetUserAgentSIP()	88
GetVersionFile()	89
GetVersionSDK()	90
SetSubjectSDP()	91
GetSubjectSDP()	92
ConfAllowLine()	93
LineVoiceChannelSPK()	94
ChatAddContact()	95
ChatSetMyStatus()	99
ForwardCall()	101
IsLineOpen()	105
DeselectAllVideoCodec()	106
SelectAllVideoCodec()	107
SelectVideoCodec()	108
DeselectVideoCodec()	109
EnableForceInfoDTMF()	110
DisableForceInfoDTMF()	111
EnableForceRfc2833DTMF()	112
DisableForceRfc2833DTMF()	113
SetLineSpkVol()	114
GetLineSpkVol()	115
AddCustomHeader()	116

RemoveCustomHeader() ..... 117  
 RemoveCustomHeaderAll() ..... 118  
 ActivateQosSIP() ..... 119  
 DeactivateQosSIP() ..... 120  
 ActivateQosRTP() ..... 121  
 DeactivateQosRTP() ..... 122  
 GetCountPacketLost() ..... 123  
 GetSizeJitterBuffer() ..... 124  
 OpenMediaSecondary() ..... 125  
 CloseMediaSecondary() ..... 126  
 ChangeMEDIA() ..... 127  
 GetVideoDevTotal() ..... 128  
 GetVideoDevName() ..... 129  
 OpenVideoDev() ..... 130  
 CloseVideoDev() ..... 131  
 CryptCOMM() ..... 132

**EXPORTED EVENTS ..... 133**

OnTryingToRegister() ..... 133  
 OnFailToRegisterEx() ..... 134  
 OnSuccessToRegister() ..... 135  
 OnTryingToUnRegister() ..... 136  
 OnFailToUnRegister() ..... 137  
 OnSuccessToUnRegister ..... 138  
 OnTryingToReRegister() ..... 139  
 OnFailToReRegisterEx() ..... 140  
 OnSuccessToReRegister ..... 141  
 OnConnecting() ..... 142  
 OnSuccessToConnect() ..... 143  
 OnFailToConnect() ..... 144  
 OnDisconnectCall() ..... 145  
 OnCallTransferAccepted() ..... 146  
 OnPlayWaveDone() ..... 147  
 OnDTMFDigit() ..... 148  
 OnMsgNOTIFY() ..... 149  
 OnIncomingCall() ..... 151  
 OnIncomingCallRingingStart() ..... 152  
 OnIncomingCallRingingStop ..... 153  
 OnProvisionalResponse() ..... 154  
 OnRedirectionResponse() ..... 155  
 OnRequestFailureResponse() ..... 156  
 OnServerFailureResponse() ..... 158  
 OnGeneralFailureResponse() ..... 159  
 OnIncomingDiagnostic() ..... 160  
 OnOutgoingDiagnostic() ..... 161  
 OnSessionLostEvent() ..... 162  
 OnSuccessToHold() ..... 163  
 OnSuccessToUnHold() ..... 166  
 OnTryingToUnHold() ..... 167  
 OnFailToUnHold() ..... 168  
 OnDetectAMD() ..... 169  
 OnChatContactStatus() ..... 170

OnChatSendMsgTextFail() .....	172
OnChatSendMsgTypingSuccess().....	173
OnChatRecvMsgText() .....	175
OnChatRecvMsgTypingStart() .....	176
OnVoiceStreamPCM() .....	178
OnFailToTransfer() .....	179
OnHoldCall().....	180
OnUnHoldCall().....	181
OnVideoRemoteShowStart() .....	182
OnVideoRemoteShowStop().....	183
OnVideoRemoteShowRGB() .....	184
OnVideoDeviceShowRGB() .....	185

## **VAXVOIP BASED SOFTPHONE CALL FLOW ..... 186**

Execution Sequence of Methods/Events to Dial a Call.....	186
Execution Sequence of Methods/Events to Receive a Call .....	187

## **INTRODUCTION AND QUICK START**

The VaxVoip SIP softphone SDK is a software development kit which is used to quickly embed SIP (Session Initiation Protocol) based softphone features to web, software and mobile phone application. It provides full support to tailor the softphones features as desired like having your own GUIs or incorporating your brand name.

## **EXPORTED FUNCTIONS**

### **InitializeEx()**

The InitializeEx() function initializes the VaxVoIP component and once the component is successfully initialized, the user will be able to dial and receive phone calls.

### **Syntax**

```
boolean InitializeEX(  
    BindtoListenIP,  
    ListenIP,  
    ListenPort,  
    UserName,  
    Login,  
    LoginPwd,  
    DisplayName,  
    DomainRealm,  
    SIPProxy,  
    SIPOutboundProxy,  
    UseSoundDevice,  
    TotalLine  
)
```

### **Parameters**

#### BindtoListenIP(boolean)

The BindtoListenIP parameter value can be 0 or 1. Assign value 1 to this parameter if you want to bind an IP address of your choice to ListenIP parameter otherwise zero.

#### ListenIP(string)

The ListenIP parameter value specifies the IP address of machine on which VaxVoIP is running. All incoming requests will be listened on this IP.

#### ListenPort(integer)

The ListenPort parameter specifies the port number for SIP softphone to receive the requests. The standard port is 5060 however any port can be dedicated for this purpose.

#### UserName(string)

This Parameter value specifies the user name which is provided by IP-Telephony service provider or VoIP providers.

#### Login(string)

This Parameter value specifies the user Login which is provided by IP-Telephony service provider or VoIP providers.

LoginPwd(string)

This Parameter value specifies the password which is provided by IP-Telephony service provider or VoIP providers.

DisplayName(string)

This Parameter value specifies the display name for user which is provided by IP-Telephony service provider or VoIP providers.

DomainRealm(string)

This Parameter value is provided by IP-Telephony service provider or VoIP providers.

SIPProxy(string)

This Parameter value is provided by IP-Telephony service provider or VoIP providers.

SIPOutboundProxy(string)

This Parameter value is provided by IP-Telephony service provider or VoIP providers.

*NOTE: In some cases, ITSP (IP-Telephony service provider) supports outbound proxy. Outbound proxy is the only way to let the NAT/firewall user to make and receive phone calls.*

*If your service provider does not provide sip outbound proxy then leave that field blank or ""*

UseSoundDevice(boolean)

The sound devices attached to the system can be captured during component initialization process by setting the value of UseSoundDevice parameter. This can be enabled/disabled by setting UseSoundDevice value 0 or 1.

TotalLine(integer)

The TotalLine parameter determines the total number of call/voice channels that can be dealt simultaneously. A specific number of lines are required to initialize the VaxVoIP component.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
InitializeEx(False, "198.168.0.103", 5060, "8002", "8002", "1234", "",  
            "sip.abc.com", "", 1, 5)  
if (Result == 0) GetVaxObjectError()
```

### See Also

UnInitialize(), GetVaxObjectError()



**UnInitialize()**

The UnInitialize() function vacates all the memory/resources that were held during component initialization.

**Syntax**

```
UnInitialize()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
UnInitialize()
```

**See Also**

InitializeEx()

## RegisterToProxy()

The RegisterToProxy() function registers the client to SIP proxy server. The registration with server is mandatory to receive calls however calls can be dialed without registration.

### Syntax

```
boolean RegisterToProxy(Expire)
```

### Parameters

Expire(integer)

The nExpire parameter specifies the time interval after which the registration with server will be refreshed consequently server will remain updated about the present client status.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
RegisterToProxy(1800)
```

### See Also

UnRegisterToProxy(), GetVaxObjectError()

**UnRegisterToProxy()**

The UnRegisterToProxy() function unregisters/disconnects the client from SIP proxy server.

**Syntax**

```
boolean UnRegisterToProxy()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
UnRegisterToProxy()
```

**See Also**

RegisterToProxy(), GetVaxObjectError()

## OpenLine()

The OpenLine() function opens a specific line to dial/receive call. As VaxVoIP supports multiple calls simultaneously so this function should be called prior to establishing connection, allowing user to dial/receive new calls on available free line.

### Syntax

```
boolean OpenLine(  
    LineNo,  
    BindtoRTPRxIP,  
    RTPRxIP,  
    RTPRxPort  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line to dial/receive call. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1

BindtoRTPRxIP(boolean)

The bBindRTPRxToListenIP parameter value can be 0 or 1(false or true). To bind a specific IP to sRTPRxIP assign value 1 to this parameter otherwise zero.

RTPRxIP(string)

The sRTPRxIP parameter value specifies the IP address of computer on which VaxVoIP receives voice streams. The ListenIP and sRTPRxIP can be different if a computer has multiple IP addresses.

RTPRxPort(int)

The sRTPRxPort parameter value specifies the port number to receive voice streams. The Listen ports should be in range of 1024 to 65535 for UDP based transmission and for RTP compliance port number should be even.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = LineOpen(2, False, "192.168.0.103", 7006)  
if (Result==0) GetVaxObjectError( )
```

### See Also

CloseLine(), GetVaxObjectError()

**CloseLine()**

The CloseLine() function closes the specific line which is no longer in use. This method can be called every time a call is disconnected to close the specific line or all open lines can be closed once at component uninitialization.

**Syntax**

```
boolean CloseLine(LineNo)
```

**Parameters**

LineNo(integer) (0 to total no of line - 1)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
LineOpen(2, False, "192.168.0.103", 7006)  
CloseLine(2)
```

**See Also**

OpenLine(), GetVaxObjectError()

**IsLineOpen()**

The IsLineOpen() function gets the OPEN status of a specific line.

**Syntax**

```
boolean IsLineOpen(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns value 1 (true) if line is already opened or 0(false) if it is closed.

**Example**

```
IsLineOpen(3)
```

**See Also**

OpenLine(), IsLineBusy()

**IsLineBusy()**

The IsLineBusy() function checks the status of already opened line i-e line is busy or free.

**Syntax**

```
boolean IsLineBusy(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns value 1 (true) if line is busy otherwise zero.

**Example**

```
IsLineBusy(4)
```

**See Also**

OpenLine(), IsLineOpen()

**IsLineConnected()**

The IsLineConnected() function checks the status of already opened line i-e line is connected or free.

**Syntax**

```
boolean IsLineConnected(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns value 1 (true) if line is connected otherwise zero.

**Example**

```
IsLineConnected(4)
```

**See Also**

OpenLine(), IsLineOpen(), IsLineBusy()



## SetLicenceKey()

The trial version of VaxVoIP SDK has trial period limitation of 30 days, so a license key is required after 30 days to avoid evaluation message box. License keys are delivered to customers on order.

The SetLicenceKey( ) method is used to make the trial version working as registered version without expiry and trial period limitation.

*NOTE: You must pay the License fee in order to get the License Key and once the License key is set, it will remove the evaluation message box & expiry.*

### Syntax

```
SetLicenceKey(LicenceKey)
```

### Parameters

LicenceKey(string)

The value of this parameter is license key provided by the company.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
SetLicenseKey("LicenseKey")
```

### See Also

Initialize(), GetVaxObjectError()

**GetVaxObjectError()**

The GetVaxObjectError() method gets the error code for the last operation which is failed to execute.

**Syntax**

```
integer GetVaxObjectError()
```

**Parameters**

No parameters

**Return Value**

The GetVaxObjectError() returns the error code.

10	VAXOBJECT is unable to initialize properly. Please call InitializeEx method to initialize VaxVoIP Object.
11	Unable to open communication port. The communication port is in use by another softphone.
12	Invalid License Key.
13	Failed to initialize VaxVoIP task window.
14	Unable to access Input device OR Input device is already in use.
15	Unable to access Output device OR Output device is already in use.
16	Microphone/Input device is not ready to use.
17	Speaker/Output device is not ready to use.
18	Unable to set the Microphone/Input device volume OR Sound device does not support microphone volume feature.
19	Unable to set Speaker/Output device volume OR Sound device does not support speaker volume feature.
20	Failed to initialize Recording media.
21	Unable to open wave file.
22	Invalid SIP URI
23	Codec is not supported.
24	Unable to create SDP (Session Description Protocol) request.
25	Unable to create CONNECTION request. Please check the provided SIP URI is valid.
26	Unable to create REGISTER request. Please check the provided SIP URI is valid.
27	Unable to create UN-REGISTER request. Please check the provided SIP URI is valid.
28	Unable to create DISCONNECT request.
29	Invalid Line Number.
30	Line is already in use.
31	Line is not open for connection.
32	Invalid Call-Id.
33	Invalid value.

34	Selected line is not in voice session.
35	Failed to read wave file.
36	Failed to write wave file.
37	Format of file is not supported.
38	Unable to create CANCEL request. Please check the provided SIP URI is valid.
39	License expired.
40	Unable to find contact OR Contact is not added.
41	Remote user is not online OR Remote user is not subscribe to the SIP SERVER
42	Error to create chat status message.
43	Error to create add contact message.
44	Unable to open/catpure video device.
45	Functionality is not supported.

### Example

```
if(Result==0)
    ErrorCode = GetVaxObjectError()
```

### See Also

InitializeEx() , SetLicenseKey()

## ConnectCall()

The ConnectCall() method creates a call request with user-defined From URI and ToURI values. In a case, if its required to send user-defined Caller-Name and Caller-Name to the IP-Telephony services provider or SIP server then ConnectCall() method can be used to achieve such functionality.

### Syntax

```
boolean ConnectCall(  
    LineNo,  
    ToURI,  
    InputDeviceId,  
    OutputDeviceId  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

FromURI(string)

The ToURI parameter value specifies To URI in SIP call request.  
*sip:username@domain/realm*  
Username in ToURI appears as caller Id.

ToURI(integer)

The ToURI parameter value specifies To URI in SIP call request.  
*sip:username@domain/realm*  
Username in FromURI appears as dial number.

InputDeviceId(integer)

This parameter specifies the id of specific input device to be connected upon call connection however -1 value can be used for default input device.

OutputDeviceId(integer)

This parameter specifies the id of specific output device to be connected upon call connection however -1 value can be used for default output device.

*NOTE: The device ID can be get using GetAudioInDevName() & GetAudioOutDevName().*

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
ConnectCall(2, "FromURIsip:7001@abc.com", -1, -1)  
ConnectCall(2, "ToURIsip:7001@abc.com", 1, 0)
```

**See Also**

DialCall(), Disconnect(), GetVaxObjectError()

## Disconnect()

The Disconnect() function disconnects the specific call in progress.

### Syntax

```
boolean Disconnect(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = Disconnect(2)  
if(Result == 0) GetVaxObjectError()
```

### See Also

DialCall(), Connect(), GetVaxObjectError()

## DialCall()

The DialCall() function dials the phone number or dials call to provided user.

*NOTE: URI for sip call request are created internally by VaxVoIP component.*

### Syntax

```
boolean DialCall(  
    LineNo,  
    DialNo,  
    InputDeviceId,  
    OutputDeviceId  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

DialNo(string)

This parameter specifies the user name or phone number to be dialed.

InputDeviceId(integer)

This parameter specifies the id of specific input device to be connected upon dialing call however -1 value can be provided for default input device.

OutputDeviceId(integer)

This parameter specifies the id of specific output device to be connected upon dialing call however -1 value can be provided for default output device.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = DialCall(2, "001914600518", -1, -1)  
if(Result == 0) GetVaxObjectError()
```

### See Also

Connect(), Disconnect(), GetAudioOutDevName(), GetAudioInDevName(), GetVaxObjectError()

**GetAudioInDevTotal()**

The GetAudioInDevTotal() function provides the total count of input devices attached to computer.

**Syntax**

```
integer GetAudioInDeviceTotal()
```

**Parameters**

No parameters.

**Return Value**

Total number of audio input devices.

**Example**

```
GetAudioInDeviceTotal()
```

**See Also**

GetAudioOutDevTotal()



**GetAudioOutDevTotal()**

The `GetAudioOutDevTotal()` function provides the total count of output devices attached to computer.

**Syntax**

```
integer GetAudioOutDeviceTotal()
```

**Parameters**

No parameters.

**Return Value**

Total number of audio output devices.

**Example**

```
GetAudioOutDeviceTotal()
```

**See Also**

`GetAudioInDevTotal()`

**GetAudioInDevName()**

The GetAudioInDevName() function returns the name of input audio device for provided device id.

**Syntax**

```
string GetAudioInDevName(DeviceId)
```

**Parameters**

DeviceId(integer)

This parameter value can be any number from zero to total number of input devices - 1. Each number corresponds to a particular audio input device.

**Return Value**

Device name for corresponding device id, otherwise empty string.

**Example**

```
GetAudioInDevName(-1)
```

**See Also**

GetAudioOutDevTotal(), GetAudioInDevTotal(), GetAudioOutDevName()

**GetAudioOutDevName()**

The GetAudioOutDevName() function returns the name of output audio device for provided device id.

**Syntax**

```
string GetAudioOutDevName(DeviceId)
```

**Parameters**

DeviceId(integer)

This parameter value can be any number from zero to total number of input devices – 1. Each number corresponds to a particular audio input device.

**Return Value**

Device name for corresponding device id, otherwise empty string.

**Example**

```
GetAudioOutDevName(0)
```

**See Also**

GetAudioInDevName(), GetAudioOutDevTotal(), GetAudioInDevTotal()

## AcceptCall()

The AcceptCall() function accepts the incoming call.

### Syntax

```
boolean AcceptCall(  
    LineNo,  
    CallId,  
    InputDeviceId,  
    OutputDeviceId  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

CallId(string)

The sCallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system (Incoming call-Id, please see OnIncomingCall() event details).

InputDeviceId(integer)

This parameter specifies the id of specific input device to be connected upon accepting call however -1 value can be provided for default input device.

OutputDeviceId(integer)

This parameter specifies the id of specific output device to be connected upon accepting call however -1 value can be provided for default output device.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = AcceptCall(1, "24c654c@192.168.0.119", 0, -1)  
if(Result == 0) GetVaxObjectError()
```

### See Also

GetAudioOutDevName(), GetAudioInDevName(), RejectCall(),  
GetVaxObjectError()

## RejectCall()

The RejectCall() function cancels/rejects the incoming call.

### Syntax

```
boolean RejectCall(CallId)
```

### Parameters

CallId(string)

The sCallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system (Incoming call-Id, please see OnIncomingCall() event details).

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = RejectCall("24c654c@192.168.0.119")  
if(Result == 0) GetVaxObjectError()
```

### See Also

AcceptCall(), GetVaxObjectError()

## TransferCallEx()

The TransferCallEx() function transfers the call from a specific line to a specific number or user. This function can be used to implement the feature “unannounced/blind call transfer i-e transferring the call without notifying the desired party/extension of the impending call”.

### Syntax

```
boolean TransferCallEx(  
    LineNo,  
    ToUserName  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1

ToUserName(string)

This parameter specifies the to user name or phone number to be dialed.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
DialCall(2, "001914600518", -1, -1)  
AcceptCall(2, "24c654c@192.168.0.119", 0, -1)  
Result = TransferCallEx(2, "00192600524")  
if(Result == 0) GetVaxObjectError()
```

### See Also

AcceptCall(), GetVaxObjectError()

## JoinTwoLine()

The JoinTwoLine() function links two calls. This function can be used to implement the feature “announced/consult call transfer i-e notifying the desired party/extension of the impending call by putting the caller on hold and dialing the desired party/extension”.

### Syntax

```
boolean JoinTwoLine(  
                    LineNoA,  
                    LineNoB  
                    )
```

### Parameters

LineNoA(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

LineNoB(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = JoinTwoLine(1, 3)  
if(Result == 0) GetVaxObjectError()
```

### See Also

TransferCallEx(), GetVaxObjectError()

**HoldLine()**

The HoldLine() method puts a specific line on hold.

**Syntax**

```
HoldLine(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = boolean HoldLine(3)
if(Result == 0) GetVaxObjectError()
```

**See Also**

HoldLine(), GetVaxObjectError()



**IsLineHold()**

The IsLineHold() method gets the HOLD status of a specific line.

**Syntax**

```
boolean IsLineHold(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns value 1 (true) if line is on hold otherwise zero.

**Example**

```
Result = IsLineHold(3)  
if(Result == 0) GetVaxObjectError()
```

**See Also**

HoldLine(), GetVaxObjectError()

**UnHoldLine()**

The UnHoldLine() function unholds a specific line.

**Syntax**

```
boolean UnHoldLine(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = UnHoldLine(2)  
if(Result == 0) GetVaxObjectError()
```

**See Also**

HoldLine(), GetVaxObjectError()

**EnableKeepAlive()**

The EnableKeepAlive() function keeps the ports open for connection by sending "keep alive packets" periodically.

It helps to keep the ports open at NAT/firewall end.

**Syntax**

```
boolean EnableKeepAlive(Seconds)
```

**Parameters**

Seconds(integer)

This nSeconds parameter value specifies the time interval after which keep alive packets will be sent to keep the port open for connection.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
OpenLine(2, False, "192.168.0.103", 7006)  
EnableKeepAlive(10)
```

**See Also**

DisableKeepAlive(), GetVaxObjectError()

**DisableKeepAlive()**

The DisableKeepAlive() method stops sending keep-alive packets i-e it disables the functionality of EnableKeepAlive method.

**Syntax**

```
void DisableKeepAlive()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
DisableKeepAlive()
```

**See Also**

EnableKeepAlive(), GetVaxObjectError()

**DeselectAllVoiceCodec()**

The DeselectAllVoiceCodec() function deselects all the voice codec options.

**Syntax**

```
void DeselectAllVoiceCodec()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
DeselectAllVoiceCodec()
```

**See Also**

SelectAllVoiceCodec(), GetVaxObjectError()

**SelectAllVoiceCodec()**

The SelectAllVoiceCodec() function selects all the voice codec options.

**Syntax**

```
void SelectAllVoiceCodec()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
SelectAllVoiceCodec()
```

**See Also**

DeselectAllVoiceCodec(), GetVaxObjectError()

## GetOutboundCodec()

The GetOutboundCodec() gets the codec number for the outbound voice stream of provided line.

### Syntax

```
integer GetOutBoundCodec(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

VaxVoIP SIP SDK support the following voice codecs:

0 = GSM 6.10

1 = iLBC

2 = G711 A-Law

3 = G711 U-Law

4 = G729

### Return Value

The function returns a codec number on its successful execution otherwise -1.

### Example

```
Result = GetOutBoundCodec(1)
if(Result == -1) ErrorMessage()
```

### See Also

GetInboundCodec(), GetVaxObjectError()

## GetInboundCodec()

The GetInboundCodec() gets the codec number for the Inbound voice stream of provided line.

### Syntax

```
integer GetInBoundCodec(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

VaxVoIP SIP SDK support the following voice codecs:

0 = GSM 6.10

1 = iLBC

2 = G711 A-Law

3 = G711 U-Law

4 = G729

### Return Value

The function returns a codec number on its successful execution otherwise -1.

### Example

```
Result = GetInBoundCodec(5)
if(Result == -1) ErrorMsg()
```

### See Also

GetOutboundCodec(), GetVaxObjectError()



## SelectVoiceCodec()

The SelectVoiceCodec() function selects a voice codec for provided codec number. The function can be called multiple times to select more than one voice codec. Moreover the sequence of selection of voice codec decides the priority of codec i-e the voice codec selected first has higher priority than the codec selected afterward.

### Syntax

```
boolean SelectVoiceCodec(CodecNo)
```

### Parameters

CodecNo(integer)

This parameter value ranges from 0-4 and each value corresponds to a particular voice codec.

VaxVoIP SIP SDK supports the following voice codecs:

0 = GSM 6.10

1 = iLBC

2 = G711 A-Law

3 = G711 U-Law

4 = G729

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
DeselectAllVoiceCodec()  
SelectVoiceCodec(0)  
SelectVoiceCodec(1)  
SelectVoiceCodec(2)  
SelectVoiceCodec(3)
```

In this example GSM6.10 has the highest priority where as G711 U-Law has lowest priority

### See Also

DeselectVoiceCodec(), GetVaxObjectError()

## DeselectVoiceCodec()

The DeselectVoiceCodec() function deselects a voice codec for provided codec number.

### Syntax

```
boolean DeselectVoiceCodec(CodecNo)
```

### Parameters

CodecNo(integer)

This parameter value ranges from 0-4 and each value corresponds to a particular voice codec.

VaxVoIP SIP SDK supports the following voice codecs:

0 = GSM 6.10

1 = iLBC

2 = G711 A-Law

3 = G711 U-Law

4 = G729

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = DeselectVoiceCodec(nCodecNo)
if(Result == 0) GetVaxObjectError()
```

### See Also

SelectVoiceCodec(), GetVaxObjectError()

**GetMyIP()**

The GetMyIP() method provides the IP address of the computer.

**Syntax**

```
string GetMyIP()
```

**Parameters**

No parameters.

**Return Value**

The function returns the IP address of the computer.

**Example**

```
GetMyIP()
```

**See Also**

GetStartMyIP(), GetNextMyIP()

**GetStartMyIP()**

The GetStartMyIP() method initiates the process to get computer IP using GetNextMyIP() method.

**Syntax**

```
boolean GetStartMyIP()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
GetStartMyIP()
```

**See Also**

GetMyIP(), GetNextMyIP()

**GetNextMyIP()**

The GetNextMyIP() method randomly gets one IP from the multiple IPs assigned to computer however it ignores already selected IP.

**Syntax**

```
string GetNextMyIP()
```

**Parameters**

No parameters.

**Return Value**

The function returns the IP address of the computer otherwise empty string.

**Example**

```
GetNextMyIP()
```

**See Also**

GetMyIP(), GetStartMyIP()

## DigitDTMF()

The DigitDTMF() function sends DTMF digit to the remote end SIP server. This method can also be used to play DTMF tones.

### Syntax

```
boolean DigitDTMF(  
                LineNo,  
                Digit  
                )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Digit(string)

This parameter value specifies any digit that has been pressed. (1, 2, 3, 4, 5, ..... 0, \*, #).

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
DigitDTMF(1, "3")
```

### See Also

SetDTMFVolume(), GetDTMFVolume()

**SetDTMFVolume()**

The SetDTMFVolume() function adjusts/sets the volume of DTMF tones.

**Syntax**

```
boolean SetDTMFVolume(Volume)
```

**Parameters**

Volume(integer)

This parameter specifies the volume level for DTMF tones ranges between 0-250.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
SetDTMFVolume(6)
```

**See Also**

DigitDTMF(), GetDTMFVolume()

**GetDTMFVolume()**

The GetDigitDTMFVolume() function returns the volume level of DTMF tones.

**Syntax**

```
integer GetDTMFVolume()
```

**Parameters**

No parameters.

**Return Value**

The function returns the volume of DTMF tones ranges between 0-250.

**Example**

```
SetDTMFVolume(6)  
GetDTMFVolume()
```

**See Also**

DigitDTMF(), SetDTMFVolume()



**EnableForceInbandDTMF()**

The EnableForceInbandDTMF() method enforces VaxVoIP component to send DTMF tones in the form of voice.

**Syntax**

```
boolean EnableForceInbandDTMF(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
EnableForceInbandDTMF(2)
```

**See Also**

DisableForceInbandDTMF( ),GetVaxObjectError()

## **DisableForceInbandDTMF()**

The `DisableForceInbandDTMF()` method disables the transmission of DTMF tones in the form of voice.

### **Syntax**

```
boolean DisableForceInbandDTMF(LineNo)
```

### **Parameters**

`LineNo`(integer)

This parameter value specifies the specific line. The `LineNo` value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### **Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling `GetVaxObjectError()` method.

### **Example**

```
DisableForceInbandDTMF(4)
```

### **See Also**

`EnableForceInbandDTMF()`, `GetVaxObjectError()`

## DetectAMD()

The DetectAMD() method enables/disables the detection of answering machine.

### Syntax

```
boolean DetectAMD (
    LineNo,
    Enable,
    AnalysisTime,
    SilenceTime,
    SilenceCount
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

Enable(boolean)

This parameter value can be 0 or 1. Assign value 1 to enable the answering machine detection on specified line or 0 to disable it.

AnalysisTime(integer)

This parameter value specifies the time interval (in millisecond )for detection of answering machine.

SilenceTime(integer)

This parameter value specifies the time interval (in millisecond) for silence i-e no human voice.

SilenceCount(integer)

This parameter value specifies the number of count for silence interval.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
DetectAMD(2, True, 6000, 300, 2)
```

### See Also

OnDetectAMD(), GetVaxObjectError()

**EnableEchoNoiseCancellation()**

The EnableEchoNoiseCancellation() enables the significant suppression of echo and any background noise. By default this is enabled to provide high quality of output speech.

**Syntax**

```
boolean EnableEchoNoiseCancellation()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
EnableEchoNoiseCancellation()
```

**See Also**

DisableEchoNoiseCancellation(), GetVaxObjectError()

**DisableEchoNoiseCancellation()**

The `DisableEchoNoiseCancellation()` disables the suppression of echo and any background noise.

**Syntax**

```
boolean DisableEchoNoiseCancellation()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling `GetVaxObjectError()` method.

**Example**

```
DisableEchoNoiseCancellation()
```

**See Also**

`EnableEchoNoiseCancellation()`, `GetVaxObjectError()`

**EnableAGC()**

The EnableAGC() function enables the automatic adjustment of speech level to a predetermined value irrespective of the user sound volume.

**Syntax**

```
boolean EnableAGC(Level)
```

**Parameters**

Level(integer)

This parameter value specifies speech level.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
EnableAGC(6)
```

**See Also**

DisableAGC(), GetVaxObjectError()

**DisableAGC()**

The DisableAGC() function disables the automatic adjustment of speech level to a predetermined value

**Syntax**

```
boolean DisableAGC()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
DisableAGC()
```

**See Also**

EnableAGC(), GetVaxObjectError()

**IsRecording()**

The IsRecording() function checks if recording is enabled or not on a specific line.

**Syntax**

```
boolean IsRecording(LineNo)
```

**Parameter:**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns value 1(true) if recording is enabled on provided line otherwise 0(false).

**Example**

```
IsRecording(6)
```

**See Also**

StartRecording(), StopRecording(), GetVaxObjectError()



## StartRecording()

The StartRecording() function starts recording voice stream on specific line.

*NOTE: VaxVoIP component creates recording tmp file for buffering purposes or to store the digital data. When this method is called, VaxVoIP component starts storing data into the tmp file.*

### Syntax

```
boolean StartRecording(  
    LineNo,  
    RecordVoice,  
    RecordCompress  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

RecordVoice(integer)

This parameter value specifies the recording mode. It can have three values and each value corresponds to a particular recording mode.

0=Record outgoing only  
1=Record incoming only  
2=Record both

RecordCompress(boolean)

The value of this parameter can be 0 or 1. Assign value 0 to this parameter to create uncompress wave file or 1 to create GSM 6.10 compress wave file.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
StartRecording(3, 1, True)  
StartRecording(1, 2, False)
```

### See Also

IsRecording(), StopRecording(), GetVaxObjectError()

**StopRecording()**

The StopRecording() function stops the recording of voice stream on specific line.

**Syntax**

```
boolean StopRecording(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
StopRecording(4)
```

**See Also**

StartRecording(), IsRecording(), GetVaxObjectError()

## ResetRecording()

The ResetRecording() method resets/clear the temporary buffer used for storing voice stream.

*NOTE: Call to this method, clears saved digital data from the recording tmp file.*

### Syntax

```
boolean ResetRecording(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
StartRecording(3, 1, True)  
ResetRecording(3)
```

### See Also

StartRecording( ), StopRecording( ), IsRecording( ), GetVaxObjectError( )

## SaveRecordingToWaveFile()

The SaveRecordingToWaveFile() saves the recorded voice data from temporary buffer at specific line to wave file.

*NOTE:* Call to this method, saves tmp voice data into wave (.wav) file.

### Syntax

```
boolean SaveRecordingToWaveFile(  
                                LineNo,  
                                FileName  
                                )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

FileName(string)

This parameter value specifies wave file name to be saved.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = SaveRecordingToWaveFile(1, "test.wav")  
if(Result == 0) GetVaxObjectError()
```

### See Also

StartRecording( ), StopRecording( ), IsRecording( ), GetVaxObjectError( ), ResetRecording()

## IsWaveFilePlaying()

The IsWaveFilePlaying() functions checks whether the wave file is in progress or not on provided line.

### Syntax

```
boolean IsWaveFilePlaying(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns value 1(true) if wave file is playing on provided line otherwise it returns 0(false).

### Example

```
IsWaveFilePlaying(2)
```

### See Also

PlayWaveOpen(), PlayWaveStart(), PlayWaveStop (), PlayWaveSkipTo(), GetVaxObjectError()

## PlayWaveOpen()

The PlayWaveOpen() function makes the wave file ready/set to play on provided line at remote end.

### Syntax

```
boolean PlayWaveOpen(  
    LineNo,  
    FileName  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

FileName(string)

This parameter value specifies wave file name to be played.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = PlayWaveOpen(6, "test.wav")  
if(Result == 0) GetVaxObjectError()
```

### See Also

IsWaveFilePlaying(), PlayWaveStart(), PlayWaveStop (), PlayWaveSkipTo(), GetVaxObjectError()

## PlayWaveClose()

The PlayWaveClose() function vacates all the resources that were held by PlayWaveOpen() function.

### Syntax

```
boolean PlayWaveClose(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
PlayWaveOpen(6, "test.wav")
Result = PlayWaveClose(6)
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayWaveOpen (), PlayWaveStart(), PlayWaveStop (), PlayWaveSkipTo(), GetVaxObjectError()

## PlayWaveStart()

The PlayWaveStart() method starts playing the already set wave file on provided line. The following sequence of execution starts playing the wave file.

- PlayWaveOpen()
- PlayWaveStart()

It starts sending wave file data to the remote end, value listen = 1 starts sending and playing (on sound card) wave file data at the same time.

### Syntax

```
boolean PlayWaveStart(  
                    LineNo,  
                    Listen  
                    )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

bListen(boolean)

This parameter value can be 0 or 1. To play wave file just to remote end set its value 0 or sets its value 1 to play wave file to both remote end and sound card.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = PlayWaveStart(6)  
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayWaveOpen (), PlayWaveClose(), PlayWaveStop (), PlayWaveSkipTo(), GetVaxObjectError()



## PlayWaveSkipTo()

The PlayWaveSkipTo() function changes the position of playing cursor to the new position.

### Syntax

```
boolean PlayWaveSkipTo(  
    LineNo,  
    Seconds  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Seconds(integer)

This parameter value specifies the time to be skipped of playing wave file.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = PlayWaveSkipTo(4, 30)  
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayWaveOpen (), PlayWaveClose(), PlayWaveStop (), PlayWaveStart(),  
GetVaxObjectError()

## **PlayWavePosition()**

The PlayWavePosition() method gets the current position of playing cursor.

### **Syntax**

```
integer PlayWavePosition(LineNo)
```

### **Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

### **Return Value**

The function returns current position of playing cursor otherwise -1.

### **Example**

```
PlayWaveOpen(4, "test.wav")
PlayWaveStart(4)
Result = PlayWavePosition(4)
if(Result == -1) ErrorMsg()
```

### **See Also**

PlayWaveOpen (), PlayWaveClose(), PlayWaveStop (), PlayWaveStart(),  
PlayWaveSkipTo(), GetVaxObjectError()

## **PlayWaveTotalTime()**

The PlayWaveTotalTime() function gets the total playing time of a wave file on provided line.

### **Syntax**

```
integer PlayWaveTotalTime(LineNo)
```

### **Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### **Return Value**

The function returns total playing time of wave file otherwise -1.

### **Example**

```
Result = PlayWaveTotalTime(4)
if(Result == -1) ErrorMsg()
```

### **See Also**

PlayWaveOpen (), PlayWaveClose(), PlayWaveStop (), PlayWaveStart(), PlayWavePause(), GetVaxObjectError()

## PlayWavePause()

The PlayWavePause() method pauses the playing wave file on its current position.

### Syntax

```
boolean PlayWavePause(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = PlayWavePause(1)  
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayWaveOpen (), PlayWaveClose(), PlayWaveStop (), PlayWaveStart(),  
PlayWaveSkipTo(), GetVaxObjectError()

## PlayWaveStop ()

The PlayWaveStop() function stops playing the wave file on provided line and position the playing cursor at the beginning of file.

### Syntax

```
boolean PalyWaveStop(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = PlayWaveStop(2)  
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayWaveOpen(), PlayWaveClose(), PlayWavePause (), PlayWaveStart(), PlayWaveSkipTo(), GetVaxObjectError()

## MuteLineSPK()

The MuteLineSPk() method mutes output voice stream of specific line.

### Syntax

```
boolean MuteLineSPK(LineNo, Enable)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Enable(boolean)

The bEnable parameter value can be 0 or 1. Assign value 1 to this parameter to mute output voice stream otherwise zero.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
MuteLineSPK(2, 0)  
MuteLineSPK(2, 1)
```

### See Also

MuteLineMIC(), GetVaxObjectError()

## MuteLineMIC()

The MuteLineMic() method mutes input voice stream of specific line.

### Syntax

```
boolean MuteLineMIC(LineNo, Enable)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Enable(boolean)

The bEnable parameter value can be 0 or 1. Assign value 1 to this parameter to mute input voice stream otherwise zero.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
MuteLineMIC(2, 0)  
MuteLineMIC(2, 1)
```

### See Also

MuteLineSPK(), GetVaxObjectError

## MuteSpk()

The MuteSpk() function mutes the speaker. Call to MuteSpk() does not affect the Master Mute Control.

### Syntax

```
boolean MuteSpk(Mute)
```

### Parameters

Mute(boolean)

The bMute parameter value can be 0 or 1. Assign value 1 to this parameter to mute the speaker otherwise zero.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
MuteSpk(0)  
MuteSpk(1)
```

### See Also

MuteMic(), GetVaxObjectError()



## MuteMic()

The MuteMic() function mutes the microphone. Call to MuteMic() method does not affect the Master Mute Control. It simply starts sending silence data.

### Syntax

```
boolean MuteMic(Mute)
```

### Parameters

Mute(boolean)

The bMute parameter value can be 0 or 1. Assign value 1 to this parameter to mute the microphone otherwise zero.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
MuteMic(0)  
MuteMic(1)
```

### See Also

MuteSpk(), GetVaxObjectError()

**GetSpkVolume( )**

The GetSpkVolume() function returns the speaker volume. The speaker volume ranges between 0-255 (0 = Min Volume, 255 = Max Volume).

**Syntax**

```
integer GetSpeakerVolume()
```

**Parameters**

No parameters.

**Return Value**

The function returns speaker volume on its successful execution otherwise -1.

**Example**

```
GetSpeakerVolume()
```

**See Also**

MuteSpk(), SetSpkVolume()

## SetSpkVolume()

The SetSpkVolume() function sets the volume of output voice stream. The speaker volume ranges between 0-255(0 = Min Volume, 255 = Max Volume).

### Syntax

```
boolean SetSpkVolume(Volume)
```

### Parameters

Volume(integer)

This parameter value specifies volume level ranges between [0-255].

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = SetSpkVolume(150)  
if(Result == 0) GetVaxObjectError()
```

### See Also

GetSpeakerVolume(), GetVaxObjectError()

**MicVolume()**

The GetMicVolume() function returns the microphone volume. The microphone volume ranges between 0-255 (0 = Min Volume, 255 = Max Volume).

**Syntax**

```
integer GetMicVolume()
```

**Parameters**

No parameters.

**Return Value**

The function returns microphone volume on its successful execution otherwise -1.

**Example**

```
GetMicVolume()
```

**See Also**

GetSpeakerVolume(), SetSpkVolume(), SetMicVolume()

## SetMicVolume()

The SetMicVolume() function sets the volume of input voice stream. The microphone volume ranges between 0-255(0 = Min Volume, 255 = Max Volume).

### Syntax

```
boolean SetMicVolume(Volume)
```

### Parameters

Volume(integer)

This parameter value specifies volume level ranges between [0-255].

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = SetMicVolume(200)
if(Result == 0) GetVaxObjectError()
```

### See Also

GetSpeakerVolume(), SetSpkVolume(), GetMicVolume()

**EnableMicBoost()**

The EnableMicBoost() method enhances the volume of input voice stream by increasing the microphone sensitivity.

**Syntax**

```
boolean EnableMicBoost()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = EnableMicBoost()  
if(Result == 0) GetVaxObjectError()
```

**See Also**

DisableMicBoost(), GetVaxObjectError()

**DisableMicBoost()**

The DisableMicBoost() disables the enhanced sensitivity of microphone.

**Syntax**

```
boolean DisableMicBoost()
```

**Parameters**

No parameters.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = DisableMicBoost()  
if(Result == 0) GetVaxObjectError()
```

**See Also**

EnableMicBoost(), GetVaxObjectError()

**IsMicBoostEnable()**

The ISMicBoostEnable() function checks the status of microphone boost i-e enabled or disabled.

**Syntax**

```
boolean IsMicBoostEnable()
```

**Parameters**

No parameters.

**Return Value**

The function returns value 1(true) if microphone boost is enabled otherwise it returns 0(false).

**Example**

```
IsMicBoostEnable()
```

**See Also**

EnableMicBoost(), DisableMicBoost(), GetVaxObjectError()



**EnableDonotDisturb()**

The EnableDonotDisturb() function blocks/prevents ringing of all incoming calls.

**Syntax**

```
void EnableDonotDsiturb()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
EnableDonotDisturb()
```

**See Also**

DisableDonotDisturb()

**DisableDonotDisturb()**

The DisableDonotDisturb() disables the functionality of EnableDonotDisturb function.

**Syntax**

```
void DisableDonotDisturb()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
DisableDonotDisturb()
```

**See Also**

EnableDonotDisturb()

**GetMicSoundLevel()**

The GetMicSoundLevel() returns the volume of microphone whereas volume ranges between 0 to 100.

**Syntax**

```
integer GetMicSoundLevel()
```

**Parameters**

No parameters.

**Return Value**

The function returns microphone volume on its successful execution otherwise -1.

**Example**

```
GetMicSoundLevel()
```

**See Also**

GetSpkSoundLevel()

**GetSpkSoundLevel()**

The GetSpkSoundLevel() returns the volume of speaker whereas speaker volume ranges between 0 to 100.

**Syntax**

```
integer GetSpkSoundLevel()
```

**Parameters**

No parameters.

**Return Value**

The function returns speaker volume on its successful execution otherwise -1.

**Example**

```
GetSpkSoundLevel()
```

**See Also**

GetMicSoundLevel()

**SetSessionLostTick()**

The SetSessionLostTick() function sets the specific time interval to check whether voice session is still intact or lost.

*NOTE: Due to some reasons, if VaxVoIP does not receives the voice stream for a specific interval of time then it triggers OnSessionLostEvent() event.*

**Syntax**

```
void SetSessionLostTicket(Minute)
```

**Parameters**

Minute(integer)

This parameter value specifies the session lost time in minutes.

**Return Value**

No return value.

**Example**

```
SetSessionLostTicket(2)
```

**See Also**

OnSessionLostEvent()

## SetSpkSoftVolume()

The SetSpkSoftVolume() function adjusts the softphone speaker volume without affecting the operating system master volume control.

### Syntax

```
boolean SetSpkSoftVolume(Volume)
```

### Parameters

Volume(integer)

This parameter value specifies volume level ranges between [0-255].

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = SetSpkSoftVolume(200)  
if(Result == 0) GetVaxObjectError()
```

### See Also

SetMicVolume(), GetMicVolume()

**SetUserAgentSIP()**

The SetUserAgentSIP() function sets the user agent field of SIP packet.

**Syntax**

```
boolean SetUserAgentSIP(UserAgentName)
```

**Parameters**

UserAgentName(string)  
This parameter value specifies the User agent Name.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = SetUserAgentSIP("abc")  
if(Result == 0) GetVaxObjectError()
```

**See Also**

GetUserAgentSIP(), GetVaxObjectError()

**GetUserAgentSIP()**

The GetUserAgentSIP() function returns the user agent field of SIP packet.

**Syntax**

```
string GetUserAgentSIP()
```

**Parameters**

No parameters.

**Return Value**

The function returns the user agent name otherwise empty string.

**Example**

```
GetUserAgentSIP()
```

**See Also**

SetUserAgentSIP()



**GetVersionFile()**

The GetVersionFile() method returns the current version of component file.

**Syntax**

```
string GetVersionFile()
```

**Parameters**

No parameters.

**Return Value**

The function returns the files/component file version number.

**Example**

```
GetVersionFile()
```

**See Also**

GetVersionSDK()

**GetVersionSDK()**

The GetVersionSDK() method returns the current version of SDK.

**Syntax**

```
string GetVersionSDK()
```

**Parameters**

No parameters.

**Return Value**

The function returns the SDK version number.

**Example**

```
GetVersionSDK()
```

**See Also**

GetVersionFile()

**SetSubjectSDP()**

The SetSubjectSDP() function sets the subject field of SIP packet.

**Syntax**

```
boolean SetSubjectSDP(SubjectSDP)
```

**Parameters**

SubjectSDP(string)

This parameter specifies the value that is to be set as subject of SIP packet.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
SetSubjectSDP("xyz")
```

**See Also**

GetSubjectSDP()

**GetSubjectSDP()**

The `GetSubjectSDP()` function returns the subject field previously set by `SetSubjectSDP()` method.

**Syntax**

```
string GetSubjectSDP()
```

**Parameters**

No parameters.

**Return Value**

The function returns the subject.

**Example**

```
GetSubjectSDP()
```

**See Also**

`SetSubjectSDP()`

## ConfAllowLine()

The ConfAllowLine() function allows multiple users to speak/listen in conference. This feature of VaxVoIP componnet can be used for supervision of operators at call centers in real time.

### Syntax

```
boolean ConfAllowLine(  
    LineNo,  
    AllowListen,  
    AllowSpeak  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

AllowListen(boolean)

This parameter value can be 0 or 1. To allow user on specific line to listen in conference sets this parameter value to 1 otherwise 0.

AllowSpeak(boolean)

This parameter value can be 0 or 1. To allow user on specific line to speak in conference sets this parameter value to 1 otherwise 0.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
ConfAllowLine(1,0,1)  
ConfAllowLine(1,0,0)  
ConfAllowLine(3,1,0)  
ConfAllowLine(3,1,1)
```

### See Also

LineVoiceChannelSPK()

## LineVoiceChannelSPK()

The LineVoiceChannelSPK() function enables/disables the right and left speaker on specific line.

### Syntax

```
boolean LineVoiceChannelSPK(  
                                LineNo,  
                                Channel  
                                )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Channel(integer)

This Parameter value specifies which speaker to be enabled /disabled.

- 0 = Enable Left Speaker
- 1 = Enable Right Speaker
- 2 = Enable both

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = LineVoiceChannelSPK(2, 1)  
if(Result == 0) GetVaxObjectError()
```

### See Also

MuteSPk(), MuteLineSPK()

## ChatAddContact()

The ChatAddContact() methods adds a contact to receive contact present status e.g online, busy, idle etc.

### Syntax

```
boolean ChatAddContact(Username)
```

### Parameters

Username(string)

This parameter value specifies the user name to be added to chat.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = ChatAddContact("abc")  
if(Result == 0) GetVaxObjectError()
```

### See Also

ChatRemoveContact(), GetVaxObjectError()

## **ChatRemoveContact()**

The ChatRemoveContact() method removes a contact that was added using ChatAddContact() method.

### **Syntax**

```
boolean ChatRemoveContact(Username)
```

### **Parameters**

Username(string)

This parameter value specifies the user name to be removed from chat.

### **Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### **Example**

```
ChatAddContact("abc")  
ChatRemoveContact("abc")
```

### **See Also**

ChatAddContact(), GetVaxObjectError()



## **ChatSendMessageTyping()**

The ChatSendMessageTyping() functions sends the typing status to remote end/user.

### **Syntax**

```
boolean ChatSendMessagingTyping(  
    UserName,  
    UserValue32bit  
)
```

### **Parameters**

UserName(string)  
This parameter value specifies the user name.

UserValue32bit(integer)  
This Parameter value is a user specified 32 bit value.

### **Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### **Example**

```
ChatSendMessagingTyping("xyz",3)
```

### **See Also**

ChatSendMessageText(), GetVaxObjectError()

## ChatSendMessageText()

The ChatSendMessageText() function sends the chat message text.

### Syntax

```
boolean ChatSendMessageText(  
    UserName,  
    MsgText,  
    MsgType,  
    UserValue32bit  
)
```

### Parameters

UserName(string)

This parameter value specifies the user name.

MsgText(string)

This parameter value specifies the message text.

MsgType(integer)

This parameter value specifies the number 101 or 102 which corresponds to particular message type.

UserValue32bit(integer)

This Parameter value is a user specified 32 bit value

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
ChatAddContact("abc")  
ChatSendMessageTyping("abc",3)  
ChatSendMessageText("abc", "xyz", 101, 3)
```

### See Also

ChatSendMessageTyping(), GetVaxObjectError()

## ChatSetMyStatus()

The ChatSetMyStatus() function sets the status of user for chat i-e online, offline, away, onphone or busy.

### Syntax

```
boolean ChatSetMyStatus(nStatusId)
```

### Parameters

StatusId(integer)

This parameter value corresponds to particular user chat status.

- 0 = online
- 1 = Offline
- 2 = Away
- 3 = On Phone
- 4 = Busy

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
ChatSetMyStatus(0)  
ChatSetMyStatus(3)
```

### See Also

ChatAddContact(), ChatRemoveContact(), ChatSendMessageText()

**VoiceChanger()**

The VoiceChanger() functions changes the pitch of the voice.

**Syntax**

```
boolean VoiceChanger(Pitch)
```

**Parameters**

Pitch(integer)

This parameter value can be -1 to disables the voice change or its value can be the pitch of the voice ranges between 0-20.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = VoiceChanger(4)  
if(Result == 0) GetVaxObjectError()
```

**See Also**

## ForwardCall()

The ForwardCall() functions forwards the call to desired user.

### Syntax

```
boolean ForwardCall(  
                    Enable,  
                    ToUserName  
                    )
```

### Parameters

Enable(boolean)

This parameter value can be 0 or 1. Assign value 1 to enable the call forwarding to particular user or 0 to disable call forwarding.

ToUserName(string)

This parameter value specifies the user name/number to be dialed.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = ForwardCall(1, "abc")  
if(Result == 0) GetVaxObjectError()
```

### See Also

Connect(), DialCall(), GetVaxObjectError()

## PlayAddPCM()

The PlayAddPCM() adds the incoming PCMs to internally created buffer of VaxVoIP component. Moreover it also plays the PCM data from buffer.

### Syntax

```
boolean PlayAddPCM(  
    LineNo,  
    DataPCM,  
    wSizePCM  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

DataPCM(string)

This parameter value specifies PCM data received from the user.

SizePCM(dword)

This parameter value specifies the size of PCM data received from the user.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = PlayAddPCM(1,"abcxyz",8)  
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayResetPCM(), CaptureStreamPCM()

**PlayResetPCM()**

The PlayResetPCM() method resets/clear VaxVoIP internally created PCM buffer.

**Syntax**

```
boolean PlayResetPCM(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = PlayResetPCM(1)  
if(Result == 0) GetVaxObjectError()
```

**See Also**

PlayAddPCM(), CaptureStreamPCM()

## CaptureStreamPCM()

The CaptureStreamPCM() function enables the process to capture incoming stream of PCM.

### Syntax

```
boolean CaptureStreamPCM(  
                                LineNo,  
                                Enable  
                                )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Enable(boolean)

This parameter value can be 0 or 1. Assign value 1 to enable the PCM data capturing on specified line or 0 to disable it.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = CaptureStreamPCM(1)  
if(Result == 0) GetVaxObjectError()
```

### See Also

PlayAddPCM(), PlayResetPCM()



**IsLineOpen()**

The IsLineOpen() function checks open/close status of a specific line.

**Syntax**

```
boolean IsLineOpen(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns value 1 (true) if line is open otherwise zero.

**Example**

```
IsLineOpen(4)
```

**See Also**

OpenLine(), IsLineBusy()

**DeselectAllVideoCodec()**

The DeselectAllVideoCodec() function deselects all the video codec options.

**Syntax**

```
void DeselectAllVideoCodec()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
DeselectAllVideoCodec()
```

**See Also**

SelectAllVideoCodec(), GetVaxObjectError()

**SelectAllVideoCodec()**

The SelectAllVideoCodec() function selects all the VaxVoIP SDK's supported video codecs.

**Syntax**

```
void SelectAllVideoCodec()
```

**Parameters**

No parameters.

**Return Value**

No return value.

**Example**

```
SelectAllVideoCodec()
```

**See Also**

DeselectAllVideoCodec(), GetVaxObjectError()

## SelectVideoCodec()

The SelectVideoCodec() function selects a video codec for provided codec number. The function can be called multiple times to select more than one video codec. Moreover the sequence of selection of video codec decides the priority of codec i-e the video codec selected first has higher priority than the codec selected afterward.

### Syntax

```
boolean SelectVideoCodec(CodecNo, Quality)
```

### Parameters

CodecNo(integer)

This parameter value ranges from 0-1 and each value corresponds to a particular video codec.

VaxVoIP SIP SDK supports the following video codecs:

0 = H263

1 = H263+

Quality(integer)

This parameter value specifies the quality.

0 = LOW

1 = STANDARD

2 = MEDIUM

3 = HIGH

4 = MAX

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
DeselectAllVideoCodec()  
SelectVideoCodec(0)  
SelectVideoCodec(1)
```

In this example H263 has the highest priority where as H263+ has lowest priority

### See Also

DeselectVideoCodec(), GetVaxObjectError()

## DeselectVideoCodec()

The DeselectVideoCodec() function deselects a video codec for provided codec number.

### Syntax

```
boolean DeselectVideoCodec(CodecNo)
```

### Parameters

CodecNo(integer)

This parameter value ranges from 0-1 and each value corresponds to a particular video codec.

VaxVoIP SIP SDK supports the following video codecs:

0 = H263

1 = H263+

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = DeselectVideoCodec(1)
if(Result == 0) GetVaxObjectError()
```

### See Also

SelectVideoCodec(), GetVaxObjectError()

## **EnableForceInfoDTMF()**

The EnableForceInfoDTMF() method enforces VaxVoIP component to send DTMF digits in the form of SIP INFO packets.

### **Syntax**

```
boolean EnableForceInfoDTMF(LineNo)
```

### **Parameters**

LineNo(integer)

This parameter value specifies the specific line. The nLineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### **Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### **Example**

```
EnableForceInfoDTMF(2)
```

### **See Also**

DisableForceRfc2833DTMF( ), GetVaxObjectError()

## DisableForceInfoDTMF()

The DisableForceInfoDTMF() method disables the transmission of DTMF digits in the form SIP INFO packets.

### Syntax

```
boolean DisableForceInfoDTMF(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The nLineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
DisableForceInfoDTMF(4)
```

### See Also

EnableForceInfoDTMF(), GetVaxObjectError()

**EnableForceRfc2833DTMF()**

The EnableForceRfc2833DTMF() method enforces VaxVoIP component to send DTMF digits in the form of RFC2833 protocol in RTP packets.

**Syntax**

```
boolean EnableForceRfc2833DTMF(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The nLineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
EnableForceRfc2833DTMF(2)
```

**See Also**

DisableForceInfoDTMF( ), GetVaxObjectError()



## **DisableForceRfc2833DTMF()**

The DisableForceRfc2833DTMF() method disables the transmission of DTMF digits in the form of RFC2833 protocol in RTP packets.

### **Syntax**

```
boolean DisableForceRfc2833DTMF(LineNo)
```

### **Parameters**

LineNo(integer)

This parameter value specifies the specific line. The nLineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### **Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### **Example**

```
DisableForceRfc2833DTMF(4)
```

### **See Also**

EnableForceRfc2833DTMF(), GetVaxObjectError()

## SetLineSpkVol()

The SetLineSpkVol() function adjusts the output volume of a specific line without affecting the operating system master volume control.

### Syntax

```
boolean SetLineSpkVol(LineNo, Volume)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The nLineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Volume(integer)

This parameter value specifies volume level ranges between [0-255].

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = SetLineSpkVol(2, 200)
if(Result == 0) GetVaxObjectError()
```

### See Also

GetLineSpkVol(), SetSpkSoftVolume()

## GetLineSpkVol()

The GetLineSpkVol() function returns the output volume of a specific line. The speaker volume ranges between 0-255 (0 = Min Volume, 255 = Max Volume).

### Syntax

```
integer GetLineSpkVol(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Return Value

The function returns speaker volume on its successful execution otherwise -1.

### Example

```
GetLineSpkVol(2)
```

### See Also

SetLineSpkVol(), SetSpkVolume()

## AddCustomHeader()

The AddCustomHeader() function can be used to add custom header fields in the SIP packets of different SIP requests.

Some of the SIP requests; REGISTER, INVITE, ACK, CANCEL, BYE, OPTIONS

### Syntax

```
boolean AddCustomHeader(ReqId, Name, Value)
```

### Parameters

ReqId (integer)

This parameter specifies a unique identification of a SIP request. Supported ReqId values are;

0 = INVITE

1 = REFER

Name (string)

This parameter specifies the name of custom header field.

Value (string)

This parameter specifies the value of custom header field.

### Return Value

On successful execution this function returns non-zero value otherwise it returns 0 value and specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = Initialize("sipsdk.com", "192.168.0.25", 5060, "192.168.0.25", -1)
if(Result = 0) GetVaxObjectError()

AddCustomHeader(0, "Call_Info", "WaitingTime = 0")
```

### See Also

RemoveCustomHeader(), RemoveCustomHeaderAll(),

## RemoveCustomHeader()

The RemoveCustomHeader() function removes the custom header fields added by using AddCustomHeader() function.

### Syntax

```
boolean RemoveCustomHeader(ReqId, Name)
```

### Parameters

ReqId (integer)

This parameter specifies a unique identification of a SIP request. Supported ReqId values are;

0 = INVITE

1 = REFER

Name (string)

This parameter specifies the custom header field.

### Return Value

On successful execution this function returns non-zero value otherwise it returns 0 value and specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
RemoveCustomHeader(0, "Call_Info")
```

### See Also

AddCustomHeader(), RemoveCustomHeaderAll()

## **RemoveCustomHeaderAll()**

The RemoveCustomHeaderAll() function removes all custom header fields added by using AddCustomHeader() function.

### **Syntax**

```
boolean RemoveCustomHeaderAll(ReqId)
```

### **Parameters**

ReqId (integer)

This parameter specifies a unique identification of a SIP request. Supported ReqId values are;

0 = INVITE

1 = REFER

### **Return Value**

On successful execution this function returns non-zero value otherwise it returns 0 value and specific error code can be retrieved by calling GetVaxObjectError() method.

### **Example**

```
RemoveCustomHeaderAll(0)
```

### **See Also**

AddCustomHeader(), RemoveCustomHeader()

## ActivateQoSIP()

The ActivateQoSIP() method activates the SIP packets priority over the communication network by adjusting the network QoS (Quality of Service) value.

### Syntax

```
boolean ActivateQoSIP(PriorityQos)
```

### Parameters

PriorityQos(integer)

This parameter specifies the SIP packets priority and adjusting the network QoS (Quality of Services) value.

- 0 = Lowest
- 1 = Lower
- 2 = Low
- 3 = High
- 4 = Higher
- 5 = Highest

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0.

### Example

```
ActivateQoSIP(0)
```

### See Also

DeactivateQoSIP()

## DeactivateQosSIP()

The DeactivateQosSIP() method deactivates the network QoS (Quality of Service) of SIP packets.

### Syntax

```
void DeactivateQosSIP()
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

- 0 = Lowest
- 1 = Lower
- 2 = Low
- 3 = High
- 4 = Higher
- 5 = Highest

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0.

### Example

```
DeactivateQosSIP(0)
```

### See Also

ActivateQosSIP()



## ActivateQosRTP()

The ActivateQosSIP() method activates the RTP packets priority over the communication network by adjusting the network QoS (Quality of Service) value.

### Syntax

```
boolean ActivateQosRTP(LineNo, PriorityQos)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

PriorityQos(integer)

This parameter specifies the RTP packets priority and adjusting the network Qos (Quality of Services) value.

- 0 = Lowest
- 1 = Lower
- 2 = Low
- 3 = High
- 4 = Higher
- 5 = Highest

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0.

### Example

```
ActivateQosRTP(0, 5)
```

### See Also

DeactivateQosRTP(), ActivateQosSIP()

**DeactivateQosRTP()**

The DeactivateQosRTP() method deactivates the network QoS (Quality of Service) of RTP packets.

**Syntax**

```
void DeactivateQosRTP(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0.

**Example**

```
DeactivateQosRTP(0)
```

**See Also**

DeactivateQosRTP(), ActivateQosSIP()

**GetCountPacketLost()**

During the call conversation, GetCountPacketLost() returns the number of UDP/RTP (voice stream) packets lost of a specific line.

**Syntax**

```
integer GetCountPacketLost(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0.

**Example**

```
Result = GetCountPacketLost(1)
if(Result == -1) GetVaxObjectError()
```

**See Also**

GetSizeJitterBuffer()

## **GetSizeJitterBuffer()**

During voice conversation, The GetCoutPacketLost() returns the size of jitter buffer of a specific line.

Jitter buffers are used to smooth delay variations in received audio by buffering the packets and adjusting their rendering. The result is a smoother delivery of audio to the user.

### **Syntax**

```
integer GetSizeJitterBuffer(LineNo)
```

### **Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

### **Return Value**

The function returns a Non-zero value on its successful execution otherwise 0.

### **Example**

```
Result = GetSizeJitterBuffer(1)
if(Result == -1) GetVaxObjectError()
```

### **See Also**

GetCountPacketLost()

## OpenMediaSecondary()

If a computer on which VaxVoIP integrated softphone is running contains more than one sound device then OpenMediaSecondary() opens and captures then secondary media device and VaxVoIP plays same voice stream on both primary and secondary sound devices.

It is very useful method and helps to develop call-center training softwares.

### Syntax

```
boolean OpenMediaSecondary(InputDeviceId, OutputDeviceId)
```

### Parameters

InputDeviceId(integer)

This parameter specifies the id of specific input device to be connected upon call connection however -1 value can be used for default input device.

OutputDeviceId(integer)

This parameter specifies the id of specific output device to be connected upon call connection however -1 value can be used for default output device.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = OpenMediaSecondary(-1, -1)  
if(Result == 0) GetVaxObjectError()
```

### See Also

CloseMediaSecondary()

**CloseMediaSecondary()**

The CloseMediaSecondary() closes the secondary media.

**Syntax**

```
boolean CloseMediaSecondary()
```

**Parameters**

No Parameters

**Return Value**

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

**Example**

```
Result = CloseMediaSecondary()  
if(Result == 0) GetVaxObjectError()
```

**See Also**

OpenMediaSecondary()

## ChangeMEDIA()

During the call session, ChangeMEDIA() method allows to shift the voice conversation from one sound device to other sound device.

### Syntax

```
boolean ChangeMEDIA(LineNo, InputDeviceId, OutputDeviceId)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

InputDeviceId(integer)

This parameter specifies the id of specific input device to be connected upon call connection however -1 value can be used for default input device.

OutputDeviceId(integer)

This parameter specifies the id of specific output device to be connected upon call connection however -1 value can be used for default output device.

*NOTE: The device ID can be get using GetAudioInDevName() & GetAudioOutDevName().*

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = ChangeMEDIA(0, -1, -1)
if(Result == 0) GetVaxObjectError()
```

### See Also

GetVideoDevTotal()

**GetVideoDevTotal()**

The GetVideoDevTotal() function provides the total number of video devices attached to the computer.

**Syntax**

```
integer GetVideoDevTotal()
```

**Parameters**

No parameters

**Return Value**

Total number of video devices.

**Example**

```
GetVideoDevTotal()
```

**See Also**

GetVideoDevName()



**GetVideoDevName()**

The GetVideoDevName() function provides the name of specific video device attached to computer.

**Syntax**

```
string GetVideoDevName(DeviceId)
```

**Parameters**

DeviceId(integer)

This parameter value can be any number from zero to total number of video devices - 1. Each number corresponds to a particular video device.

**Return Value**

Device name for corresponding device id, otherwise empty string.

**Example**

```
GetVideoDevName(-1)
```

**See Also**

GetVideoDevTotal()

## OpenVideoDev()

The OpenVideoDev() function opens a specific video device attached to computer.

### Syntax

```
boolean OpenVideoDev(DeviceId, Quality)
```

### Parameters

DeviceId(integer)

This parameter value can be any number from zero to total number of video devices - 1. Each number corresponds to a particular video device.

Quality(integer)

This parameter value specifies the quality.

0 = LOW

1 = STANDARD

2 = MEDIUM

3 = HIGH

4 = MAX

### Return Value

The function returns value 1(true) if wave file is playing on provided line otherwise it returns 0(false).

### Example

```
Result = OpenVideoDev(0, 3)  
if(Result == 0) GetVaxObjectError()
```

### See Also

CloseVideoDev()

**CloseVideoDev()**

The CloseVideoDev() function closes a specific video device previously opened by using OpenVideoDev() function.

**Syntax**

```
void CloseVideoDev(DeviceId)
```

**Parameters**

DeviceId(integer)

This parameter value can be any number from zero to total number of video devices - 1. Each number corresponds to a particular video device.

**Return Value**

No return value

**Example**

```
CloseVideoDev(0)
```

**See Also**

OpenVideoDev()

## CryptCOMM()

The CryptCOMM() function makes encrypt the communication data.

### Syntax

```
boolean CryptCOMM(  
    Enable  
    RemoteIP  
    RemotePort  
)
```

### Parameters

Enable(boolean)  
This parameter value enables/disables crypted communication through VaxVoIP Tunneling Server.

RemoteIP(string)  
This parameter value specifies the IP address of the computer on which VaxVoIP Tunneling Server is running.

RemotePort(integer)  
This parameter value specifies the listen port number of the computer on which VaxVoIP Tunneling Server is running.

### Return Value

The function returns a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

### Example

```
Result = CryptCOMM(1, "66.77.88.99", 8891)  
if(Result == 0) GetVaxObjectError()
```

## **EXPORTED EVENTS**

### **OnTryingToRegister()**

VaxVoIP triggers OnTryingToRegister() event when client sends the register request to SIP server and request is in process at server end.

#### **Syntax**

```
void OnTryingToRegister()
```

#### **Parameters**

No parameters.

#### **Example**

```
OnTryingToRegister()  
{  
}  
}
```

#### **See Also**

OnTryingToUnRegister(), OnFailToRegisterEx(), OnSuccessToRegister(), RegisterToProxy(), UnRegisterToProxy()

## OnFailToRegisterEx()

The OnFailToRegisterEx() event triggers when client failed to register with server or registration request has not completed successfully.

### Syntax

```
void OnFailToRegisterEx(StatusCode, ReasonPhrase)
```

### Parameters

StatusCode(integer)

This parameter specifies SIP response status code (486, 404 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Unauthorized, Not Found etc).

### Example

```
OnFailToRegisterEx()  
{  
}
```

### See Also

OnFailToUnRegister(), OnFailToRegisterEx(), OnSuccessToRegister(), RegisterToProxy(), UnRegisterToProxy()

**OnSuccessToRegister()**

The OnSuccessToRegister() event triggers when client successfully registered with SIP server.

**Syntax**

```
void OnSuccessToRegister()
```

**Parameters**

No parameters.

**Example**

```
OnSuccessToRegister  
{  
}
```

**See Also**

OnTryingToRegister(), OnFailToRegisterEx(), OnTryingToUnRegister()  
RegisterToProxy(), UnRegisterToProxy()

**OnTryingToUnRegister()**

The OnTryingToUnRegister() event triggers when client sends the unregister request to SIP server and request is in process at server end.

**Syntax**

```
void OnTryingToUnRegister()
```

**Parameters**

No parameters.

**Example**

```
OnTryingToUnRegister()  
{  
}  
}
```

**See Also**

OnTryingToRegister(), OnFailToRegisterEx(), OnSuccessToRegister()  
RegisterToProxy(), UnRegisterToProxy()



**OnFailToUnRegister()**

The OnFailToUnRegister() event triggers when client failed to unregister with server or unregister request has not been completed successfully.

**Syntax**

```
void OnFailToUnRegister()
```

**Parameters**

No parameters.

**Example**

```
OnFailToUnRegister()  
{  
}  
}
```

**See Also**

OnSuccessToUnRegister(), OnSuccessToRegister(), OnTryingToUnRegister()  
RegisterToProxy(), UnRegisterToProxy()

**OnSuccessToUnRegister**

The OnSuccessToUnRegister() events triggers when client request to unregister with server is successfully completed.

**Syntax**

```
void OnSuccessToUnRegister()
```

**Parameters**

No parameters.

**Example**

```
OnSuccessToUnRegister()  
{  
}  
}
```

**See Also**

OnFailToUnRegister(), OnSuccessToRegister(), OnTryingToUnRegister()  
RegisterToProxy(), UnRegisterToProxy()

**OnTryingToReRegister()**

OnTryingToReRegister() event triggers when client sends re-register request to SIP server and request is in process at server end.

It notifies that sip server is processing the re-register request.

**Syntax**

```
void OnTryingToReRegister()
```

**Parameters**

No parameters.

**Example**

```
OnTryingToReRegister()  
{  
}  
}
```

**See Also**

OnSuccessToReRegister(), OnFailToReRegister(), RegisterToProxy(),  
UnRegisterToProxy()

## OnFailToReRegisterEx()

The OnFailToReRegisterEx() event triggers when client failed to re-register with server or re-registration request has not completed successfully.

### Syntax

```
void OnFailToReRegisterEx(StatusCode, ReasonPhrase)
```

### Parameters

StatusCode(integer)

This parameter specifies SIP response status code (486, 404 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Unauthorized, Not Found etc).

### Example

```
OnFailToReRegisterEx()  
{  
  
}
```

### See Also

OnTryingToReRegister(), OnSuccessToReRegister(), RegisterToProxy(), UnRegisterToProxy()

**OnSuccessToReRegister**

The OnSuccessToReRegister() event triggers when client successfully re-registered with SIP server.

**Syntax**

```
void OnSuccessToReRegister()
```

**Parameters**

No parameters.

**Example**

```
OnSuccessToRegister()  
{  
}  
}
```

**See Also**

OnTryingToReRegister(), OnFailToReRegister(), RegisterToProxy(),  
UnRegisterToProxy()

## OnConnecting()

The OnConnecting() event triggers when client dials a number on specific line.

### Syntax

```
void OnConnecting(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Example

```
OnConnecting(LineNo)  
{  
}  
}
```

### See Also

OnSuccessToConnect(), OnFailToConnect(), Connect(), Disconnect()

## OnSuccessToConnect()

The OnSuccessToConnect() event triggers when a connection is successfully established between the two parties.

### Syntax

```
void OnSuccessToConnect(  
                        LineNo,  
                        ToRTPIP,  
                        ToRTPPort  
                        )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

ToRTPIP(string)

This parameter specifies the RTP IP address of remote end.

ToRTPPort(integer)

This parameter specifies the RTP port number of remote end.

### Example

```
OnSuccessToConnect(LineNo, sToRTPIP, nToRTPPort)  
{  
    StopDialTone()  
    GetSpkVolume()  
    GetMicVolume()  
}
```

### See Also

OnFailToConnect(), OnDisconnectCall(), Connect(), Disconnect()

**OnFailToConnect()**

The OnFailToConnect() event triggers when time out occurs at client side i-e client did not receive any response from SIP server for specific interval of time.

**Syntax**

```
void OnFailToConnect(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnFailToConnect()  
{  
}  
}
```

**See Also**

OnSuccessToConnect(), OnDisconnectCall(), Connect(), Disconnect()



**OnDisconnectCall()**

The OnDisconnectCall() event triggers when remote party hang up the phone.

**Syntax**

```
void OnDisconnectCall(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnDisconnectCall(LineNo)
{
    StopDialTone()
}
```

**See Also**

OnSuccessToConnect(), OnFailToConnect(), Connect(), Disconnect()

**OnCallTransferAccepted()**

The OnCallTransferAccepted() event triggers when SIP server acknowledged/accepted the call transfer request.

**Syntax**

```
void OnCallTransferAccepted(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnCallTransferAccepted(LineNo)  
{  
}  
}
```

**See Also**

TransferCallEx(), JoinTwoLine()

**OnPlayWaveDone()**

The OnPlayWaveDone() event triggers when entire wave file has been played by the component on specific line.

**Syntax**

```
void OnPlayWaveDone(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnPlayWaveDone(LineNo)
{
}
```

**See Also**

PlayWaveOpen(), PlayWaveClose(), PlayWaveStart(), PlayWaveStop()

## OnDTMFDigit()

The OnDTMFDigit() event triggers when remote end pressed any key/DTMF.

### Syntax

```
void OnDTMFDigit(  
                LineNo,  
                Digit  
                )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

Digit(string)

This parameter value specifies any digit that has been pressed.  
(1, 2, 3, 4, 5, ..... 0, \*, #)

### Example

```
OnDTMFDigit(LineNo, sDigit)  
{  
}
```

### See Also

DigitDTMF(), SetDTMFVolume(), GetDTMFVolume

**OnMsgNOTIFY()**

The OnMsgNOTIFY() event triggers when client/softphone receives NOTIFY message from the SIP server.

**Syntax**

```
void OnMsgNOTIFY(Msg)
```

**Parameters**

Msg(string)  
This parameter specifies complete SIP request data.

**Example**

```
OnMsgNOTIFY(sMsg)  
{  
}
```

**See Also**

OnVoiceMailMsg()

## OnVoiceMailMsg()

The OnVoiceMailMsg() event triggers when client get voice mail notification from SIP server. This event only works if voice mail message service is enabled in SIP server.

### Syntax

```
void OnVoiceMailMsg(  
    IsMsgWaiting,  
    NewMsgCount,  
    NewUrgentMsgCount,  
    OldUrgentMsgCount,  
    MsgAccount  
)
```

### Parameters

IsMsgWaiting(boolean)  
This parameter value specifies whether some message is in waiting state or not.

NewMsgCount(integer)  
This parameter specifies total count for new messages.

NewUrgentMsgCount(integer)  
This parameter value specifies total count for new urgent messages.

OldUrgentMsgCount(integer)  
This parameter value specifies total count for old urgent messages.

MsgAccount(string)  
This parameter value specifies message account.

### Example

```
OnVoiceMailMsg(bIsMsgWaiting, dwNewMsgCount, dwNewUrgentMsgCount,  
    OldUrgentMsgCount, MsgAccount)  
{  
}
```

### See Also

OnMsgNOTIFY()

## OnIncomingCall()

The OnIncomingCall() event triggers when component gets incoming call.

### Syntax

```
void OnInComingCall(  
    CallId,  
    DisplayName,  
    UserName,  
    FromURI,  
    ToURI  
)
```

### Parameters

CallId(string)

The sCallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

DisplayName(string)

This Parameter value is provided by IP-Telephony service provider or VoIP providers.

UserName(string)

This Parameter value specifies the user name which is provided by IP-Telephony service provider or VoIP providers.

FromURI(string)

This parameter specifies FromURI in incoming SIP call request.

ToURI(string)

This parameter specifies ToURI in incoming SIP call request.

### Example

```
OnInComingCall(sCallId, DisplayName, UserName, sFromURI, ToURI)  
{  
}
```

### See Also

AcceptCall(), RejectCall(), HoldLine()

## OnIncomingCallRingingStart()

The OnIncomingCallRingingStart() event triggers when Client gets incoming call from remote user. Any phone bell wave file can be played on this event.

### Syntax

```
void OnIncomingCallRingingStart(sCallId)
```

### Parameters

CallId(string)

The sCallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

### Example

```
OnIncomingCallRingingStart(sCallId)
{
    StartTone()
}
```

### See Also

AcceptCall(), RejectCall(), HoldLine()



## OnIncomingCallRingingStop

The OnIncomingCallRingingStop() event triggers when remote end cancels the call. This event stops playing phone bell wave file.

### Syntax

```
void OnIncomingCallRingingStop(sCallId)
```

### Parameters

CallId(string)

The sCallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

### Example

```
OnIncomingCallRingingStop(sCallId)
{
    StopTone()
}
```

### See Also

AcceptCall(), RejectCall(), HoldLine()

## OnProvisionalResponse()

The OnProvisionalResponse() event triggers when client dials a phone call and receives provision response from SIP server. The SIP provisional responses lie in the range of 1xx (100 to 199). Please see the SIP RFC 3261 for more details.

SIP Provisional responses 1xx			
100	Trying	180	Ringing
181	Call Is Being Forwarded	182	Queued
183	Session Progress		

## Syntax

```
void OnProvisionalResponse(  
    LineNo,  
    StatusCode,  
    ReasonPhrase  
)
```

## Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

StatusCode(integer)

This parameter specifies SIP response status code (100, 181 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Trying, Ringing etc).

## Example

```
OnProvisionalResponse(LineNo, StatusCode, ReasonPhrase)  
{  
}
```

## See Also

DialCall(), Connect(), OnRedirectionResponse(), OnRequestFailureResponse()

## OnRedirectionResponse()

The OnRedirectionResponse() event triggers when client dials a phone call and receives redirection response from SIP server. The SIP redirection responses lie in the range of 3xx (300 to 399). Please see the SIP RFC 3261 for more details.

Redirection 3xx			
300	Multiple Choices	301	Moved Permanently
302	Moved Temporarily	305	Use Proxy
380	Alternative Service		

## Syntax

```
void OnRedirectionResponse(  
    StatusCode,  
    ReasonPhrase,  
    Contact  
)
```

## Parameters

StatusCode(integer)

This parameter specifies SIP response status code (300, 380 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Trying, Ringing etc).

Contact(string)

This parameter value specifies the contact where SIP server will redirect the call.

## Example

```
OnRedirectionResponse( nStatusCode, sReasonPhrase, sContact)  
{  
}
```

## See Also

Disconnect(), OnProvisionResponse(), OnRequestFailureResponse()

**OnRequestFailureResponse()**

The OnRequestFailureResponse() event triggers when client dials a phone call and receives request failure response from SIP server. The SIP request failure responses lie in the range of 4xx (400 to 499). Please see the SIP RFC 3261 for more details.

Request Failure 4xx			
400	Bad Request	401	Unauthorized
402	Payment Required	403	Forbidden
404	Not Found	405	Method Not Allowed
406	Not Acceptable	407	Proxy Authentication Required
408	Request Timeout	410	Gone
413	Request Entity Too Large	414	Request-URI Too Long
415	Unsupported Media Type	416	Unsupported URI Scheme
420	Bad Extension	421	Extension Required
423	Interval Too Brief	480	Temporarily Unavailable
481	Call/Transaction Does Not Exist	482	Loop Detected
483	Too Many Hops	484	Address Incomplete
485	Ambiguous	486	Busy Here
487	Request Terminated	488	Not Acceptable Here
491	Request Pending	493	Undecipherable

**Syntax**

```
void OnRequestFailureResponse(
    LineNo,
    StatusCode,
    ReasonPhrase
)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

StatusCode(integer)

This parameter specifies SIP response status code (486, 423 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Unauthorized, Not Found etc).

**Example**

```
OnRequestFailureResponse(LineNo, StatusCode, ReasonPhrase)
{
}
```

**See Also**

Disconnect(), OnProvisionResponse(), OnRedirectionResponse()

## OnServerFailureResponse()

The OnServerFailureResponse() event triggers when client dials a phone call and receives server failure response from SIP server. The SIP server failure responses lie in the range of 5xx (500 to 599). Please see the SIP RFC 3261 for more details.

Server Failure 5xx			
500	Server Internal Error	501	Not Implemented
502	Bad Gateway	503	Service Unavailable
504	Server Time-out	505	Version Not Supported
513	Message Too Large		

## Syntax

```
void OnServerFailureResponse(  
    LineNo,  
    StatusCode,  
    ReasonPhrase  
)
```

## Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

StatusCode(integer)

This parameter specifies SIP response status code (504, 505 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Bad Gateway, Service Unavailable etc).

## Example

```
OnServerFailureResponse(LineNo, StatusCode, ReasonPhrase)  
{  
}  
}
```

## See Also

OnProvisionResponse(), OnRedirectionResponse(), RequestFailureResponse()

## OnGeneralFailureResponse()

The OnGeneralFailureResponse() event triggers when client dials a phone call and receives global failure response from SIP server. The SIP general failure responses lie in the range of 6xx (600 to 699). Please see the SIP RFC 3261 for more details.

Global Failures 6xx			
600	Busy Everywhere	603	Decline
604	Does Not Exist Anywhere	606	Not Acceptable

## Syntax

```
void OnGeneralFailureResponse(  
    LineNo,  
    StatusCode,  
    ReasonPhrase  
)
```

## Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

StatusCode(integer)

This parameter specifies SIP response status code (600, 606 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Decline, Not Accepted etc).

## Example

```
OnGeneralFailureResponse(LineNo, StatusCode, ReasonPhrase)  
{  
}
```

## See Also

OnProvisionResponse(), OnRedirectionResponse(), RequestFailureResponse(), OnServerFailureResponse()

## OnIncomingDiagnostic()

The OnIncomingDiagnostic() event triggers when VaxVoIP receives a SIP packet. This event can be use for logging and monitoring of inbound SIP messages.

### Syntax

```
void OnIncomingDiagnostic(  
    MsgSIP,  
    FromIP,  
    FromPort  
)
```

### Parameters

- MsgSIP(string)  
This parameter value specifies the SIP packet message.
- FromIP(string)  
This parameter value specifies the *from* IP address.
- FromPort(integer)  
This parameter specifies the *from* port number.

### Example

```
OnIncomingDiagnostic(MsgSIP, FromIP, FromPort)  
{  
}
```

### See Also

OnOutgoingDiagnostic()



## OnOutgoingDiagnostic()

The OnOutgoingDiagnostic() event triggers when VaxVoIP sends a SIP packet. This event can be use for logging and monitoring of outbound SIP messages.

### Syntax

```
void OnIncomingDiagnostic(  
    MsgSIP,  
    ToIP,  
    ToPort  
)
```

### Parameters

MsgSIP(string)  
This parameter value specifies the SIP packet message.

ToIP(string)  
This parameter value specifies the to IP address.

ToPort(integer)  
This parameter specifies the to port number.

### Example

```
OnOutgoingDiagnostic(MsgSIP, ToIP, ToPort)  
{  
}
```

### See Also

OnIncomingDiagnostic()

**OnSessionLostEvent()**

The OnSessionLostEvent() triggers only when client has already enabled session lost through SetSessionLostTick() and has not received any voice data for specified interval of time.

**Syntax**

```
void OnSessionLostEvent(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnSessionLostEvent(LineNo)
{
}
```

**See Also**

SetSessionLostTick()

**OnSuccessToHold()**

The OnSuccessToHold() event triggers when a call is successfully placed on hold.

**Syntax**

```
void OnSuccessToHold(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

**Example**

```
OnSuccessToHold(LineNo)
{
}
```

**See Also**

OnTryingToHold(), OnFailToHold(), HoldLine(), UnHoldLine(), IsLineHold()

**OnTryingToHold()**

The OnTryingToHold() event triggers when client sends the hold request for specific line to SIP server and request is in process at server end.

**Syntax**

```
void OnTryingToHold(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnTryingToHold(LineNo)  
{  
}
```

**See Also**

OnSuccessToHold(), OnFailToHold(), HoldLine(), UnHoldLine(), IsLineHold()

**OnFailToHold()**

The OnFailToHold() event triggers when hold request to server has not been completed successfully.

**Syntax**

```
void OnFailToHold(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnFailToHold(LineNo)  
{  
}
```

**See Also**

OnSuccessToHold(), OnTryingToHold(), HoldLine(), UnHoldLine(),  
IsLineHold().

**OnSuccessToUnHold()**

The OnSuccessToUnHold() event triggers when request to unhold a specific line is completed successfully.

**Syntax**

```
void OnSuccessToUnHold(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnSuccessToUnHold(LineNo)
{
}
```

**See Also**

OnTryingToUnHold(), OnFailToUnHold(), HoldLine(), UnHoldLine(),  
IsLineHold().

**OnTryingToUnHold()**

The OnTryingToUnHold() event triggers when client sends the unhold request for specific line to SIP server and request is in process at server end.

**Syntax**

```
void OnTryingToUnHold(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
OnTryingToUnHold(LineNo)
{
}
```

**See Also**

OnSuccessToUnHold(), OnFailToUnHold(), HoldLine(), UnHoldLine(), IsLineHold().

## OnFailToUnHold()

The OnFailToUnHold() event triggers when unhold request to server has not been completed successfully.

### Syntax

```
void OnFailToUnHold(LineNo)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

### Example

```
OnFailToUnHold(LineNo)  
{  
}
```

### See Also

OnTryingToUnHold(), OnSuccessToUnHold(), HoldLine(), UnHoldLine(), IsLineHold().



## OnDetectAMD()

The OnDetectAMD() event triggers when request for detection of answering machine on specific line is successfully completed.

### Syntax

```
void OnDectecAMD(  
                LineNo,  
                IsHuman  
                )
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

bIsHuman(boolean)

This parameter value can be 0 or 1. The value 1 corresponds to human voice and value 0 corresponds to answering machine.

### Example

```
void OnDetectAMD(LineNo, IsHuman)  
{  
}  
}
```

### See Also

DetectAMD()

## OnChatContactStatus()

The OnChatContactStatus() event triggers when remote party/user changes the status e.g. busy, away etc.

### Syntax

```
void OnChatContactStatus(  
                        UserName,  
                        StatusId  
                        )
```

### Parameters

UserName(string)

This parameter value specifies the user name.

StatusId(integer)

This parameter value corresponds to particular user chat status.

0 = online

1 = Offline

2 = Away

3 = On Phone

4 = Busy

### Example

```
void OnChatContactStatus(UserName, StatusId)  
{  
}  
}
```

### See Also

ChatSetMyStatus(), ChatAddContact()

## OnChatSendMsgTextSuccess()

The OnChatSendMsgTextSuccess() event triggers when message is sent successfully.

### Syntax

```
void OnChatSendMsgTextSuccess(  
    UserName,  
    MsgText,  
    UserValue32bit  
)
```

### Parameters

UserName(string)  
This parameter value specifies the user name.

MsgText(string)  
This parameter value specifies the message text.

UserValue32bit(integer)  
This parameter value is a user specified 32 bit value.

### Example

```
void OnChatSendMsgTextSuccess(UserName, MsgText, UserValue32bit)  
{  
}  
}
```

### See Also

OnChatSendMsgTextFail(), ChatSendMessageText()

## OnChatSendMsgTextFail()

The OnChatSenMsgTextFail() event triggers when message sending to remote end failed.

### Syntax

```
void OnChatSendMsgTextFail(  
    UserName,  
    StatusCode,  
    ReasonPhrase,  
    MsgText,  
    UserValue32bit  
)
```

### Parameters

UserName(string)

This parameter value specifies the user name.

StatusCode(integer)

This parameter specifies SIP response status code.

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Trying, Ringing etc).

MsgText(string)

This parameter value specifies the message text.

UserValue32bit(integer)

This parameter value is a user specified 32 bit value.

### Example

```
void OnChatSendMsgTextFail(UserName, nStatusCode, sReasonPhrase,  
    MsgText, UserValue32bit)  
{  
}  
}
```

### See Also

OnChatSendMsgTextSuccess(), ChatSendMessageText()

## OnChatSendMsgTypingSuccess()

The OnChatSendMsgTypingSuccess() event triggers when typing message is sent successfully.

### Syntax

```
void OnChatSendMsgTypingSuccess(  
                                UserName,  
                                UserValue32bit  
                                )
```

### Parameters

UserName(string)  
This parameter value specifies the user name.

UserValue32bit(integer)  
This parameter value is a user specified 32 bit value.

### Example

```
void OnChatSendMsgTypingSuccess(UserName, UserValue32bit)  
{  
}  
}
```

### See Also

OnChatSendMsgTypingFail(), ChatSendMessageTyping()

## OnChatSendMsgTypingFail()

The OnChatSenMsgTypingFail() event triggers when typing message sending to remote end failed.

### Syntax

```
void OnChatSendMsgTypingFail(  
    UserName,  
    StatusCode,  
    ReasonPhrase,  
    UserValue32bit  
)
```

### Parameters

UserName(string)

This parameter value specifies the user name.

StatusCode(integer)

This parameter specifies SIP response status code.

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Trying, Ringing etc).

UserValue32bit(integer)

This parameter value is a user specified 32 bit value.

### Example

```
void OnChatSendMsgTypingFail(UserName, StatusCode, ReasonPhrase,  
    UserValue32bit)  
{  
}
```

### See Also

OnChatSendMsgTypingSuccess(), ChatSendMessageTyping()

## OnChatRecvMsgText()

The OnChatRecvMsgText() event triggers when VaxVoIP component received a text message.

### Syntax

```
void OnChatRecvMsgText(  
    UserName,  
    MsgText  
)
```

### Parameters

UserName(string)  
This parameter value specifies the user name.

MsgText(string)  
This parameter value specifies the message text.

### Example

```
OnChatRecvMsgText (UserName, MsgText)  
{  
}
```

### See Also

OnChatSendMsgTextSuccess(), ChatSendMessageText()

## **OnChatRecvMsgTypingStart()**

The OnChatRecvMsgTypingStart() event triggers when a user at remote end starts typing a text message.

### **Syntax**

```
void OnChatRecvMsgTypingStart(Username)
```

### **Parameters**

Username(string)  
This parameter value specifies the user name.

### **Example**

```
OnChatRecvMsgTypingStart(Username)  
{  
}
```

### **See Also**

OnChatSendMsgTypingFail(), ChatSendMessageTyping(),  
OnChatSendMsgTypingSuccess(), ChatSendMessageTyping()



## **OnChatRecvMsgTypingStop()**

The OnChatRecvMsgTypingStop() event triggers when a user at remote end stops typing a text message.

### **Syntax**

```
void OnChatRecvMsgTypingStop(Username)
```

### **Parameters**

Username(string)  
This parameter value specifies the user name.

### **Example**

```
OnChatRecvMsgTypingStop(Username)  
{  
}
```

### **See Also**

OnChatSendMsgTypingSuccess(), ChatSendMessageTyping(),  
OnChatSendMsgTypingFail(), ChatSendMessageTyping()

## OnVoiceStreamPCM()

The OnVoiceStreamPCM() event triggers when VaxVoIP component gets the incoming voice stream PCM on specific line.

### Syntax

```
void OnVoiceStreamPCM(  
    LineNo,  
    DataPCM,  
    SizePCM  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

DataPCM(string)

This parameter value specifies PCM data received from the user.

SizePCM(integer)

This parameter value specifies the size of PCM data received from the user.

### Example

```
void OnVoiceStreamPCM(LineNo, DataPCM, SizePCM)  
{  
}
```

### See Also

CaptureStreamPCM(), PlayAddPCM(), PlayResetPCM()

## OnFailToTransfer()

The OnFailToTransfer() event triggers when call transfer process fails and SIP Server sends an error response.

### Syntax

```
void OnFailToTransfer(LineNo, StatusCode, ReasonPhrase)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

StatusCode(integer)

This parameter specifies SIP response status code (486, 404 etc).

ReasonPhrase(string)

This parameter specifies SIP response reason phrase (Unauthorized, Not Found etc).

### Example

```
void OnFailToTransfer(LineNo, StatusCode, ReasonPhrase)
{
}
}
```

### See Also

TransferCallEx(), JoinTwoLine()

**OnHoldCall()**

The OnHoldCall() event triggers if VaxVoIP component receives hold request from the SIP Server.

**Syntax**

```
void OnHoldCall(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
void OnHoldCall(LineNo)
{
}
}
```

**See Also**

OnUnHoldCall()

**OnUnHoldCall()**

The OnUnHoldCall() event triggers if VaxVoIP component receives unhold request from the SIP Server.

**Syntax**

```
void OnUnHoldCall(LineNo)
```

**Parameters**

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

**Example**

```
void OnUnHoldCall(LineNo)
{
}
}
```

**See Also**

OnHoldCall()

## OnVideoRemoteShowStart()

The `OVideoRemoteShowStart()` event triggers when VaxVoIP component starts receiving the video frames.

### Syntax

```
void OnVideoRemoteShowStart(LineNo)
```

### Parameters

`LineNo`(integer)

This parameter value specifies the specific line. The `LineNo` value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

### Example

```
void OnVideoRemoteShowStart(LineNo)
{
    //Initialize DirectX or Screen component.
}
```

### See Also

`OnVideoRemoteShowStop()`, `OVideoRemoteShowRGB()`

## OnVideoRemoteShowStop()

The `OVideoRemoteShowStop()` event triggers when VaxVoIP component stops receiving the video frames.

### Syntax

```
void OnVideoRemoteShowStop(LineNo)
```

### Parameters

`LineNo(integer)`

This parameter value specifies the specific line. The `LineNo` value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines - 1.

### Example

```
void OnVideoRemoteShowStop(LineNo)
{
    //UnInitialize DirectX or Screen component.
}
```

### See Also

`OnVideoRemoteShowStart()`, `OVideoRemoteShowRGB()`

## OnVideoRemoteShowRGB()

The OVideoRemoteShowRGB() event triggers when VaxVoIP component receives a video frame.

### Syntax

```
void OnVideoRemoteShowRGB(  
    LineNo  
    FrameRGB  
    FrameSize  
    FrameWidth  
    FrameHeight  
)
```

### Parameters

LineNo(integer)

This parameter value specifies the specific line. The LineNo value is a unique number to identify a specific line. The range of line number is between 0 to Total number of lines – 1.

FrameRGB(integer)

Frame data in the form of RGB. VaxVoIP provided VaxSIPUserAgentShowOCX.dll can be used to show the data, please see sample code (VBNet, cSharp or Delphi) for further details.

FrameSize(integer)

Specifies the Frame data size.

FrameWidth(integer)

Specifies the Frame width.

FrameHeight(integer)

Specifies the Frame height.

### Example

```
void OnVideoRemoteShowRGB (LineNo, FrameRGB, FrameSize,  
    FrameWidth, FrameHeight)  
{  
    //Show frame using VaxSIPUserAgentShowOCX.dll  
    //Show frame using directX  
}
```

### See Also

OVideoDeviceShowRGB()



## OnVideoDeviceShowRGB()

The OVideoDeviceShowRGB() event triggers when VaxVoIP component receives a video frame from camera device for preview purposes.

### Syntax

```
void OVideoDeviceShowRGB(  
    DeviceId  
    FrameRGB  
    FrameSize  
    FrameWidth  
    FrameHeight  
)
```

### Parameters

DeviceId(integer)

This parameter value can be any number from zero to total number of video devices – 1. Each number corresponds to a particular video device.

FrameRGB(integer)

Frame data in the form of RGB. VaxVoIP provided VaxSIPUserAgentShowOCX.dll can be used to show the data, please see sample code (VBNet, cSharp or Delphi) for further details.

FrameSize(integer)

Specifies the Frame data size.

FrameWidth(integer)

Specifies the Frame width.

FrameHeight(integer)

Specifies the Frame height.

### Example

```
void OVideoDeviceShowRGB(DeviceId, FrameRGB, FrameSize,  
    FrameWidth, FrameHeight)  
{  
    //Show frame using VaxSIPUserAgentShowOCX.dll  
    //Show frame using directX  
}
```

### See Also

OnVideoRemoteShowRGB(), OpenVideoDev()

## **VaxVoIP based Softphone Call Flow**

Softphones developed via VaxVoIP SDK can easily make and receive SIP (Session Initiation Protocol) based phone calls through any SIP gateway or SIP based IP-Telephony service provider. The softphone employs methods of VaxVoIP component in certain sequence to dial and receive phone calls. Following is the execution sequence of methods and events to dial and receive calls through VaxVoIP based softphone.

### **Execution Sequence of Methods/Events to Dial a Call**

Method:InitializeEx()
Method:OpenLine()
Method:RegisterToProxy()
Event: OnTryingToRegister()
Event: OnSuccessToRegister()
VaxVoIP component successfully registered with SIP server.
Method: DialCall()/Connect()
Event: OnConnecting()
Event: OnProvisionResponse()
Remote party/User accepts the call.
Event: OnSuccessToConnect()
Call/Voice session successfully establish.
Method:DisConnect()/OnDisconnectCall()
Remote Party/User hang up the phone.

**Execution Sequence of Methods/Events to Receive a Call**

Method:InitializeEx()
Method:OpenLine()
Method:RegisterToProxy()
Event: OnTryingToRegister()
Event: OnSuccessToRegister()
VaxVoIP component successfully registered with SIP server.
Event: OnIncomingCall()
Event : OnIncomingCallRingingStart()
Method: AcceptCall()
Remote party/User receives the call.
Event: OnSuccessToConnect()
Call/Voice session successfully establish.
Method:DisConnect()/OnDisconnectCall()
Remote Party/User hang up the phone.