VaxVoIP
SIP SOFTPHONE SDK

# SIP SOFTPHONE SDK

## Microsoft Windows Desktop OS
## TECHNICAL DOCUMENTATION
## VERSION 1.2

# CONTENTS

# 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(
                     bBindtoListenIP,
                     sListenIP,
                     nListenPort,
                     sUserName,
                     sLogin,
                     sLoginPwd,
                     sDisplayName,
                     sDomainRealm,
                     sSIPProxy,
                     sSIPOutboundProxy,
                     bUseSoundDevice,
                     nTotalLine
                     )
```

### Parameters

bBindToListenIP(boolean)

The bBindToListenIP 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 sListenIP parameter otherwise zero.

sListenIP(string)

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

nListenPort(integer)

The nListenport 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.

sUserName(string)

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

sLogin(string)

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

sLoginPwd(string)

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

sDisplayName(string)

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

sDomainRealm(string)

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

sSIPProxy(string)

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

sSIPOutBoundProxy(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 ""*

bUseSoundDevice(boolean)

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

nTotalLine(integer)

The nTotalLine 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(nExpire)
```

**Parameters**

nExpire(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(
                 nLineNo,
                 bBindtoRTPRxIP,
                 sRTPRxIP,
                 nRTPRxPort
                )
```

### Parameters

nLineNo(integer)
This parameter value specifies the specific line to dial/receive call. 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

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

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

nRTPRxPort(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(nLineNo)
```

**Parameters**

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

> 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**

```
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(nLineNo)
```

### Parameters

nLineNo(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 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(nLineNo)
```

**Parameters**

nLineNo(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 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(nLineNo)
```

### Parameters

nLineNo(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 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(sLicenceKey)
```

**Parameters**

sLicenceKey(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. |

**Example**

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

**See Also**

Initialize() , SetLicenseKey()

## Connect()

The Connect() function connects the call at provided sip URI.

> *NOTE: For a number to be dialed a URI is required to create for sip call request*

### Syntax

```
boolean Connect(
                  nLineNo,
                  sToURI,
                  nInputDeviceId,
                  nOutputDeviceId
                 )
```

### Parameters

nLineNo(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.

sToURI(string)

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

nInputDeviceId(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.

nOutputDeviceId(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

```
Connect(2, ”sip:8002@abc.com”, -1, -1)
Connect(2, ”sip:8002@abc.com”, 1, 0)
```

**See Also**

DialCall(), DisConnect(), GetVaxObjectError()

## Disconnect()

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

### Syntax

```
boolean Disconnect(nLineNo)
```

### Parameters

nLineNo(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

```
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(
                 nLineNo,
                 sDialNo,
                 nInputDeviceId,
                 nOutputDeviceId
                )
```

### Parameters

nLineNo(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.

sDialNo(string)

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

nInputDeviceId(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.

nOutputDeviceId(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() functions gets the name of input audio device for provided device id.

### Syntax

```
string GetAudioInDevName(nDeviceId)
```

### Parameters

nDeviceId(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() functions gets the name of output audio device for provided device id.

**Syntax**

```
string GetAudioOutDevName(nDeviceId)
```

**Parameters**

nDeviceId(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(
                    nLineNo,
                    sCallId,
                    nInputDeviceId,
                    nOutputDeviceId
                   )
```

**Parameters**

nLineNo(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.

sCallId(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).

nInputDeviceId(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.

nOutputDeviceId(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(sCallId)
```

### Parameters

sCallId(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(
                      nLineNo,
                      sToUserName
                     )
```

### Parameters

nLineNo(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

sToUserName(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(
                     nLineNoA,
                     nLineNoB
                    )
```

**Parameters**

nLineNoA(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.

nLineNoB(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**

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

**See Also**

TransferCallEx(), GetVaxObjectError()

## HoldLine()

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

### Syntax

```
HoldLine(nLineNo)
```

### Parameters

nLineNo(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

```
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(nLineNo)
```

### Parameters

nLineNo(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 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(nLineNo)
```

### Parameters

nLineNo(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

```
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(nSeconds)
```

### Parameters

nSeconds(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(nLineNo)
```

#### Parameters

nLineNo(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.

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) ErrorMsg()
```

#### See Also

GetInboundCodec(), GetVaxObjectError()

### GetInboundCodec()

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

**Syntax**

```
integer GetInBoundCodec(nLineNo)
```

**Parameters**

nLineNo(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.

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(nCodecNo)
```

### Parameters

nCodecNo(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(nCodecNo)
```

### Parameters

nCodecNo(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(
                    nLineNo,
                     sDigit
                    )
```

#### Parameters

nLineNo(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.

sDigit(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(nVolume)
```

#### Parameters

nVolume(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(nLineNo)
```

### Parameters

nLineNo(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

```
EnableForceInbandDTMF(2)
```

### See Also

DisableForceInbandDTMF( ),GetVaxObjectError()

### DisableForceInbandDTMF()

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

**Syntax**

```
boolean DisableForceInbandDTMF(nLineNo)
```

**Parameters**

nLineNo(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**

```
DisableForceInbandDTMF(4)
```

**See Also**

EnableForceInbandDTMF(), GetVaxObjectError()

## DetectAMD()

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

### Syntax

```
boolean DetectAMD (
                        nLineNo,
                        bEnable,
                        nAnalysisTime,
                        nSilenceTime,
                        nSilenceCount
                      )
```

### Parameters

nLineNo(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.

bEnable(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.

nAnalysisTime(integer)

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

nSilenceTime(integer)

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

nSilenceCount(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(nLevel)
```

### Parameters

nLevel(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(nLineNo)
```

### Parameter:

nLineNo(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 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 /channel.

*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(
                         nLineNo,
                         nRecordVoice,
                         bRecordCompress
                        )
```

### Parameters

nLineNo(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.

nRecordVoice(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

bRecordCompress(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 /channel.

### Syntax

```
boolean StopRecording(nLineNo)
```

### Parameters

nLineNo(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

```
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(nLineNo)
```

### Parameters

nLineNo(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

```
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(
                                 nLineNo,
                                 sFileName
                               )
```

### Parameters

nLineNo(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.

sFileName(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(nLineNo)
```

### Parameters

nLineNo(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 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(
                      nLineNo,
                      sFileName
                     )
```

**Parameters**

nLineNo(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.

sFileName(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(nLineNo)
```

### Parameters

nLineNo(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

```
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(
                        nLineNo,
                        bListen
                       )
```

### Parameters

nLineNo(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.

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 PlaySkipTo(
                    nLineNo,
                    nSeconds
                   )
```

### Parameters

nLineNo(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.

nSeconds(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 = PlaySkipTo(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(nLineNo)
```

#### Parameters

nLineNo(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 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(nLineNo)
```

### Parameters

nLineNo(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 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(nLineNo)
```

**Parameters**

nLineNo(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**

```
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(nLineNo)
```

**Parameters**

nLineNo(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**

```
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(
                          nLineNo,
                          bEnable
                        )
```

**Parameters**

nLineNo(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.

bEnable(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(
                        nLineNo,
                        bEnable
                       )
```

**Parameters**

nLineNo(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.

bEnable(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(bMute)
```

### Parameters

bMute(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(bMute)
```

#### Parameters

bMute(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(nVolume)
```

#### Parameters

nVolume(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(nVolume)
```

**Parameters**

nVolume(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(nMinute)
```

### Parameters

nMinute(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(nVolume)
```

### Parameters

nVolume(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(sUserAgentName)
```

#### Parameters

sUserAgentName(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 e.g: 7,0,8,4.

### Syntax

```
string GetVersionFile()
```

### Parameters

No parameters.

### Return Value

The function returns the files/component 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(sSubjectSDP)
```

### Parameters

sSubjectSDP(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(
                         nLineNo,
                         bAllowListen,
                         bAllowSpeak
                       )
```

### Parameters

nLineNo(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.

bAllowListen(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.

bAllowSpeak(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(
                              nLineNo,
                              nChannel
                            )
```

#### Parameters

nLineNo(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.

nChannel(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(sUserName)
```

**Parameters**

sUserName(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(sUserName)
```

### Parameters

sUserName(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(
                                 sUserName,
                                 nUserValue32bit
                                )
```

**Parameters**

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

nUserValue32bit(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(
                             sUserName,
                             sMsgText,
                             sMsgType,
                             nUserValue32bit
                            )
```

**Parameters**

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

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

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

nUserValue32bit(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”)
ChatSendMessagingTyping(“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**

nStatusId(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(nPitch)
```

**Parameters**

nPitch(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(
                    bEnable,
                    sToUserName
                   )
```

### Parameters

bEnable(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.

sToUserName(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(
                        nLineNo,
                        pDataPCM,
                        dwSizePCM
                      )
```

**Parameters**

nLineNo(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.

pDataPCM(string)
This parameter value specifies PCM data received from the user.

dwSizePCM(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(nLineNo)
```

#### Parameters

nLineNo(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

```
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(
                              nLineNo,
                              bEnable
                            )
```

### Parameters

nLineNo(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.

bEnable(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()

# 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(), OnFailToRegister(), OnSuccessToRegister(), RegisterToProxy(), UnRegisterToProxy()

## OnFailToRegister()

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

### Syntax

```
void OnFailToRegister()
```

### Parameters

No parameters.

### Example

```
OnFailToRegister()
{
}
```

### See Also

OnFailToUnRegister(), OnFailToRegister(), 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(), OnFailToRegister(), 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(), OnFailToRegister(), 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()

## OnFailToReRegister()

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

### Syntax

```
void OnFailToReRegister()
```

### Parameters

No parameters.

### Example

```
OnFailToReRegister()
{
}
```

### 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(nLineNo)
```

### Parameters

nLineNo(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.

### Example

```
OnConnecting(nLineNo)
{
}
```

### See Also

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

### OnSuccessToConnect()

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

**Syntax**

```
void OnSuccessToConnect(
                         nLineNo,
                         sToRTPIP,
                         nToRTPPort
                        )
```

**Parameters**

nLineNo(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.

sToRTPIP(string)
This parameter specifies the RTP IP address of remote end.

nToRTPPort(integer)
This parameter specifies the RTP port number of remote end.

**Example**

```
OnSuccessToConnect(nLineNo, 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(nLineNo)
```

**Parameters**

nLineNo(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.

**Example**

```
OnFailToConnect()
{
}
```

**See Also**

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

## OnDisconnectCall()

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

### Syntax

```
void OnDisconnectCall(nLineNo)
```

### Parameters

nLineNo(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.

### Example

```
OnDisconnectCall(nLineNo)
{
        StopDialTone()
}
```

### See Also

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

### OnCallTransferAccepted()

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

**Syntax**

```
void OnCallTransferAccepted(nLineNo)
```

**Parameters**

nLineNo(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.

**Example**

```
OnCallTransferAccepted(nLineNo)
{
}
```

**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(nLineNo)
```

**Parameters**

nLineNo(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.

**Example**

```
OnPlayWaveDone(nLineNo)
{
}
```

**See Also**

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

### OnDTMFDigit()

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

**Syntax**

```
void OnDTMFDigit(
                   nLineNo,
                   sDigit
                  )
```

**Parameters**

nLineNo(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.

sDigit(string)

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

**Example**

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

**See Also**

DigitDTMF(), SetDTMFVolume(), GetDTMFVolume

### OnMsgNOTIFY()

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

**Syntax**

```
void OnMsgNOTIFY(sMsg)
```

**Parameters**

sMsg(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(
                        bIsMsgWaiting,
                        dwNewMsgCount,
                      dwNewUrgentMsgCount,
                        dwOldUrgentMsgCount,
                      sMsgAccount
                        )
```

**Parameters**

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

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

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

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

sMsgAccount(string)
This parameter value specifies message account.

**Example**

```
OnVoiceMailMsg(bIsMsgWaiting, dwNewMsgCount, dwNewUrgentMsgCount,
                   dwOldUrgentMsgCount, sMsgAccount)
{
}
```

**See Also**

OnMsgNOTIFY(()

### OnIncomingCall()

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

#### Syntax

```
void OnInComingCall(
                    sCallId,
                    sDisplayName,
                    sUserName,
                    sFromURI,
                    sToURI
                   )
```

#### Parameters

sCallId(string)

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

sDisplayName(string)

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

sUserName(string)

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

sFromURI(string)

This parameter specifies FromURI in incoming SIP call request.

sToURI(string)

This parameter specifies ToURI in incoming SIP call request.

#### Example

```
OnInComingCall(sCallId, sDisplayName, sUserName, sFromURI, sToURI)
{
}
```

#### 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**

sCallId(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**

sCallId(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 OnProvisionResponse() 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(
                                          nLineNo,
                                           nStatusCode,
                                           sReasonPhrase
                                          )
```

### Parameters

nLineNo(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.

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

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

### Example

```
        OnProvisionalResponse(nLineNo, nStatusCode, sReasonPhrase)
        {
        }
```

### 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(
                               nStatusCode,
                               sReasonPhrase,
                               sContact
                              )
```

### Parameters

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

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

sContact(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(
                              nLineNo,
                              nStatusCode,
                              sReasonPhrase
                             )
```

**Parameters**

nLineNo(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.

nStatusCode(integer)

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

sReasonPhrase(string)

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

### Example

```
OnRequestFailureResponse(nLineNo, nStatusCode, sReasonPhrase)
{
}
```

### 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(
                                        nLineNo,
                                        nStatusCode,
                                        sReasonPhrase
                                       )
```

### Parameters

nLineNo(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.

nStatusCode(integer)

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

sReasonPhrase(string)

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

### Example

```
OnServerFailureResponse(nLineNo, nStatusCode, sReasonPhrase)
{
}
```

### 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(
                                    nLineNo,
                                    nStatusCode,
                                    sReasonPhrase
                                   )
```

### Parameters

nLineNo(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.

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

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

### Example

```
OnGeneralFailureResponse(nLineNo, nStatusCode, sReasonPhrase)
{
}
```

### 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(
                         sMsgSIP,
                         sFromIP,
                         nFromPort
                        )
```

### Parameters

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

sFromIP(string)
This parameter value specifies the *from* IP address.

nFromPort(integer)
This parameter specifies the *from* port number.

### Example

```
OnIncomingDiagnostic(sMsgSIP, sFromIP, nFromPort)
{
}
```

### 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(
                             sMsgSIP,
                             sToIP,
                             nToPort
                            )
```

### Parameters

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

sToIP(string)
This parameter value specifies the t*o* IP address.

nToPort(integer)
This parameter specifies the *to* port number.

### Example

```
OnOutgoingDiagnostic(sMsgSIP, sToIP, nToPort)
{
}
```

### 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(nLineNo)
```

**Parameters**

nLineNo(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.

**Example**

```
OnSessionLostEvent(nLineNo)
{
}
```

**See Also**

SetSessionLostTick()

## OnSuccessToHold()

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

### Syntax

```
void OnSuccessToHold(nLineNo)
```

### Parameters

nLineNo(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.

### Example

```
OnSuccessToHold(nLineNo)
{
}
```

### 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(nLineNo)
```

#### Parameters

nLineNo(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.

#### Example

```
OnTryingToHold(nLineNo)
{
}
```

#### 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(nLineNo)
```

### Parameters

nLineNo(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.

### Example

```
OnFailToHold(nLineNo)
{
}
```

### 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(nLineNo)
```

**Parameters**

nLineNo(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.

**Example**

```
OnSuccessToUnHold(nLineNo)
{
}
```

**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(nLineNo)
```

### Parameters

nLineNo(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.

### Example

```
OnTryingToUnHold(nLineNo)
{
}
```

### 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(nLineNo)
```

#### Parameters

nLineNo(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.

#### Example

```
OnFailToUnHold(nLineNo)
{
}
```

#### 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(
                 nLineNo,
                 bIsHuman
                )
```

**Parameters**

nLineNo(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.

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(nLineNo, bIsHuman)
{
}
```

**See Also**

DetectAMD()

## OnChatContactStatus()

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

### Syntax

```
void OnChatContactStatus(
                         sUserName,
                         nStatusId
                        )
```

### Parameters

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

nStatusId(integer)
This parameter value corresponds to particular user chat status.

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

### Example

```
void OnChatContactStatus(sUserName, nStatusId)
{
}
```

### See Also

ChatSetMyStatus(), ChatAddContact()

## OnChatSendMsgTextSuccess()

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

### Syntax

```
void OnChatSendMsgTextSuccess(
                                          sUserName,
                                          sMsgText,
                                          nUserValue32bit
                                         )
```

### Parameters

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

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

nUserValue32bit(integer)
This parameter value is a user specified 32 bit value.

### Example

```
void OnChatSendMsgTextSuccess(sUserName, sMsgText, nUserValue32bit)
{
}
```

### See Also

OnChatSendMsgTextFail(), ChatSendMessageText()

## OnChatSendMsgTextFail()

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

### Syntax

```
    void OnChatSendMsgTextFail(
                               sUserName,
                               nStatusCode,
                               sReasonPhrase,
                               sMsgText,
                               nUserValue32bit
                              )
```

### Parameters

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

nStatusCode(integer)
This parameter specifies SIP response status code.

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

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

nUserValue32bit(integer)
This parameter value is a user specified 32 bit value.

### Example

```
    void OnChatSendMsgTextFail(sUserName, nStatusCode, sReasonPhrase,
                               sMsgText, nUserValue32bit)
    {
    }
```

### See Also

OnChatSendMsgTextSuccess(), ChatSendMessageText()

### OnChatSendMsgTypingSuccess()

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

**Syntax**

```
void OnChatSendMsgTypingSuccess(
                                sUserName,
                                nUserValue32bit
                               )
```

**Parameters**

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

nUserValue32bit(integer)
This parameter value is a user specified 32 bit value.

**Example**

```
void OnChatSendMsgTypingSuccess(sUserName, nUserValue32bit)
{
}
```

**See Also**

OnChatSendMsgTypingFail(), ChatSendMessageTyping()

## OnChatSendMsgTypingFail()

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

### Syntax

```
void OnChatSendMsgTypingFail(
                                  sUserName,
                                  nStatusCode,
                                  sReasonPhrase,
                                  nUserValue32bit
                                 )
```

### Parameters

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

nStatusCode(integer)
This parameter specifies SIP response status code.

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

nUserValue32bit(integer)
This parameter value is a user specified 32 bit value.

### Example

```
void OnChatSendMsgTypingFail(sUserName, nStatusCode, sReasonPhrase,
                                   nUserValue32bit)
{
}
```

### See Also

OnChatSendMsgTypingSuccess(), ChatSendMessageTyping()

### OnChatRecvMsgText()

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

**Syntax**

```
void OnChatRecvMsgText(
                        sUserName,
                        sMsgText
                       )
```

**Parameters**

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

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

**Example**

```
OnChatRecvMsgText (sUserName, sMsgText)
{
}
```

**See Also**

OnChatSendMsgTextSuccess(), ChatSendMessageText()

### OnChatRecvMsgTypingStart()

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

**Syntax**

```
void OnChatRecvMsgTypingStart(sUserName)
```

**Parameters**

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

**Example**

```
OnChatRecvMsgTypingStart(sUserName)
{
}
```

**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(sUserName)
```

### Parameters

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

### Example

```
OnChatRecvMsgTypingStop(sUserName)
{
}
```

### 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(
                        nLineNo,
                        pDataPCM,
                        nSizePCM
                       )
```

### Parameters

nLineNo(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.

pDataPCM(string)
This parameter value specifies PCM data received from the user.

nSizePCM(integer)
This parameter value specifies the size of PCM data received from the user.

### Example

```
void OnVoiceStreamPCM(nLineNo, pDataPCM, nSizePCM)
{
}
```

### See Also

CaptureStreamPCM(), PlayAddPCM(), PlayResetPCM()

# 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. |