

VaxVoIP

SIP SOFTPHONE SDK

SIP SOFTPHONE SDK
Apple MAC Desktop OS
TECHNICAL DOCUMENTATION
VERSION 1.4

CONTENTS

INTRODUCTION AND QUICK START 4

EXPORTED FUNCTIONS..... 5

InitializeEx() 5
 RegisterToProxy() 8
 UnRegisterToProxy() 9
 DialCall()10
 Connect()11
 Disconnect()12
 AcceptCall()13
 RejectCall()14
 JoinTwoLine()15
 TransferCallEx()16
 HoldLine()17
 UnHoldLine()18
 IsLineConnected()19
 IsLineBusy()20
 IsLineHold()22
 OpenLine()23
 DigitDTMF()25
 DeselectAllVoiceCodec()26
 SelectAllVoiceCodec()27
 SelectVoiceCodec()28
 DeselectVoiceCodec()29
 EnableKeepAlive()30
 DisableKeepAlive()31
 SetLicenceKey()32
 GetMyIP()33
 GetVaxObjectError()34
 MuteLineSPK()36
 MuteLineMIC()37
 DonotDisturb()39

EXPORTED EVENTS 40

OnSuccessToRegister()40
 OnSuccessToReRegister41
 OnSuccessToUnRegister42
 OnTryingToRegister()43
 OnTryingToReRegister()44
 OnTryingToUnRegister()45
 OnFailToRegister()46
 OnFailToReRegister()47
 OnFailToUnRegister()48
 OnConnecting()49
 OnTryingToHold()50
 OnTryingToUnHold()51
 OnFailToHold()52
 OnFailToUnHold()53

OnSuccessToHold()	54
OnSuccessToUnHold()	55
OnFailToConnect()	56
OnIncomingCall()	57
OnIncomingCallRingingStart()	58
OnIncomingCallRingingStop	59
OnConnected()	60
OnProvisionalResponse()	61
OnFailureResponse()	62
OnRedirectResponse()	64
OnDisconnectCall()	65
OnCallTransferAccepted()	66
OnFailToTransfer()	67

VAXVOIP BASED SOFTPHONE CALL FLOW 68

Execution Sequence of Methods/Events to Dial a Call	68
Execution Sequence of Methods/Events to Receive a Call	69

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,  
    pListenIP,  
    nListenPort,  
    pUserName,  
    pLogin,  
    pLoginPwd,  
    pDisplayName,  
    pDomainRealm,  
    pSIPProxy,  
    pSIPOutboundProxy,  
    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.

pListenIP(string)

The pListenIP 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.

pUserName(string)

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

pLogin(string)

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

pLoginPwd(string)

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

pDisplayName(string)

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

pDomainRealm(string)

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

pSIPProxy(string)

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

pSIPOutBoundProxy(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 ""

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", "", 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()

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,  
    pDialNo,  
    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.

pDialNo(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(), GetVaxObjectError()

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,  
    pToURI,  
    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.

pToURI(string)

The sToURI parameter value specifies To URI in SIP call request.
sip:username@domain/realms
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.

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()

AcceptCall()

The AcceptCall() function accepts the incoming call.

Syntax

```
boolean AcceptCall(  
    nLineNo,  
    pCallId,  
    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.

pCallId(string)

The pCallId 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", -1, -1)  
if(Result == 0) GetVaxObjectError()
```

See Also

RejectCall(), GetVaxObjectError()

RejectCall()

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

Syntax

```
boolean RejectCall(pCallId)
```

Parameters

pCallId(string)

The pCallId 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()

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()

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,  
    pToUserName  
)
```

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

pToUserName(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", -1, -1)  
Result = TransferCallEx(2, "00192600524")  
if(Result == 0) GetVaxObjectError()
```

See Also

AcceptCall(), 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 = HoldLine(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()

IsLineConnected()

The IsLineConnected() function get the status of line i-e 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 a Non-zero value on its successful execution otherwise 0, a specific error code can be retrieved by calling GetVaxObjectError() method.

Example

```
Result = IsLineConnected(2)
```

See Also

GetVaxObjectError()

IsLineBusy()

The IsLineBusy() function checks the status of already opened line i-e line is connected 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

```
Result= IsLineBusy(4)
```

See Also

OpenLine(), IsLineOpen()

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

```
Result = IsLineOpen(3)
```

See Also

OpenLine(), IsLineBusy()

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)
```

See Also

HoldLine(), 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,  
    pRTPRxIP,  
    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 true). To bind a specific IP to sRTPRxIP assign value 1 to this parameter otherwise zero.

pRTPRxIP(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()

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,  
                nDigit  
                )
```

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.

nDigit(integer)

This parameter value specifies any digit that has been pressed.

(1, 2, 3, 4, 5.....0) # is equivalent to 11 and * is equivalent to 10.

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, 10)
```

See Also

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(), SelectVoiceCodec(), DeselectVoiceCodec()

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(), DeselectVoiceCodec, SelectVoiceCodec()

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 0-3 and each value corresponds to a particular voice codec.

VaxVoIP SIP SDK supports the following voice codecs:

0 = G711 U-Law

1 = G711 A-Law

2 = GSM 6.10

3 = iLBC

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 G711U-Law has the highest priority where as iLBC has lowest priority.

See Also

DeselectVoiceCodec(), SelectAllVoiceCodec(), DeSelectAllVoiceCodec()

DeselectVoiceCodec()

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

Syntax

```
boolean DeselectVoiceCodec(nCodecNo)
```

Parameters

nCodecNo(integer)

This parameter value ranges from 0-3 and each value corresponds to a particular voice codec.

VaxVoIP SIP SDK supports the following voice codecs:

0 = G711 U-Law

1 = G711 A-Law

2 = GSM 6.10

3 = iLBC

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(), SelectAllVoiceCodec(), DeSelectAllVoiceCodec()

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 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()

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(pLicenceKey)
```

Parameters

pLicenceKey(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

InitializeEx(), 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

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	Please make your SIP account online.
11	Local communication port is currently busy, please re-try!
12	License key is invalid!
13	Failed to initialize task manager, please re-try!
14	Unable to access sound input device, please re-try!
15	Unable to access sound output device, please re-try!
16	Unable to access sound input device, please re-try!
17	Unable to access sound output device, please re-try!
18	Your sound device does not support microphone volume.
19	Your sound device does not support speaker volume.
20	Recording media initialization failed!
21	Unable to initialize recording process.
22	Provided SIP URI is invalid.
23	Selected voice codec is not supported.
24	Error to create SDP packet, please re-try!
25	Error to create CONNECTION packet, please re-try!
26	Error to create SIP REGISTER packet, please re-try!
27	Error to create SIP UN-REGISTER packet, please re-try!
28	Error to create SIP DISCONNECT packet, please re-try!
29	Selected Phone-Line is invalid!
30	Selected Phone-Line is currently busy!
31	Selected Phone-Line is not ready to use!
32	Invalid Call-ID!
33	Invalid argument(s)!
34	Selected line is not in voice session!
35	Failed to read sound file
36	Failed to write sound file
37	Unsupported sound file format.
38	Error to create SIP CANCEL packet, please re-try!
39	License expired.

Example

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

See Also

Initialize() , SetLicenseKey()

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

EchoCancellation()

The EchoCancellation() method enables/disables the suppression of echo. By default this is enabled to provide high quality of output speech.

Syntax

```
boolean EchoCancellation(bEnable)
```

Parameters

bEnable(boolean)

The bEnable parameter value can be 0 or 1. Assign value 1 to this parameter to enable echo cancellation otherwise zero to disable echo cancellation.

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

```
EchoCancellation(1)
```

See Also

GetVaxObjectError()

DonotDisturb()

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

Syntax

```
boolean DonotDsiturb(bEnable)
```

Parameters

bEnable(boolean)

The bEnable parameter value can be 0 or 1. Assign value 1 to this parameter to block all incoming calls 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

```
DonotDisturb(bEnable)
```

See Also

EXPORTED EVENTS

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()

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()

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()

OnTryingToRegister()

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

Syntax

```
void OnTryingToRegister()
```

Parameters

No parameters.

Example

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

See Also

OnTryingToUnRegister(), OnFailToRegister(), OnSuccessToRegister(), RegisterToProxy(), UnRegisterToProxy()

OnTryingToReRegister()

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

OnTryingToUnRegister()

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

Syntax

```
void OnTryingToUnRegister()
```

Parameters

No parameters.

Example

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

See Also

OnTryingToRegister(), OnFailToRegister(), OnSuccessToRegister()
RegisterToProxy(), UnRegisterToProxy()

OnFailToRegister()

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

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 OnFailToRegister(  
    nStatusCode,  
    pReasonPhrase  
)
```

Parameters

nStatusCode(integer)

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

pReasonPhrase(string)

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

Example

```
OnFailToRegister(nStatusCode, pReasonPhrase)  
{  
}
```

See Also

OnFailToUnRegister(), OnFailToRegister(), OnSuccessToRegister(), 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(  
    nStatusCode,  
    pReasonPhrase  
)
```

Parameters

nStatusCode(integer)

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

pReasonPhrase(string)

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

Example

```
OnFailToReRegister(nStatusCode, pReasonPhrase)  
{  
}
```

See Also

OnTryingToReRegister(), OnSuccessToReRegister(), 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(  
                        nStatusCode,  
                        pReasonPhrase  
                        )
```

Parameters

nStatusCode(integer)

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

pReasonPhrase(string)

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

Example

```
OnFailToUnRegister(nStatusCode, pReasonPhrase)  
{  
}
```

See Also

nSuccessToUnRegister(), OnSuccessToRegister(), OnTryingToUnRegister()
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()

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()

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().

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().

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().

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()

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().

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()

OnIncomingCall()

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

Syntax

```
void OnInComingCall(  
    pCallId,  
    pDisplayName,  
    pUserName,  
    pFromURI,  
    pToURI  
)
```

Parameters

pCallId(string)

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

pDisplayName(string)

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

pUserName(string)

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

pFromURI(string)

This parameter specifies FromURI in incoming SIP call request.

pToURI(string)

This parameter specifies ToURI in incoming SIP call request.

Example

```
OnInComingCall(pCallId, pDisplayName, pUserName, pFromURI, pToURI)  
{  
}
```

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(pCallId)
```

Parameters

pCallId(string)

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

Example

```
OnIncomingCallRingingStart(pCallId)
{
}
```

See Also

AcceptCall(), RejectCall(), HoldLine()

OnIncomingCallRinginStop

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

Syntax

```
void OnIncomingCallRinginStop(pCallId)
```

Parameters

pCallId(string)

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

Example

```
OnIncomingCallRinginStop(pCallId)  
{  
}
```

See Also

AcceptCall(), RejectCall(), HoldLine()

OnConnected()

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

Syntax

```
void onConnected(  
    nLineNo,  
    pTxRTPIP,  
    nTxRTPPort,  
    pCallId  
)
```

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.

pTxRTPIP(string)

This parameter value specifies the IP address of computer on which VaxVoIP receives voice streams.

nTxRTPPort(integer)

This 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.

pCallId(string)

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

Example

```
OnConnected(nLineNo, pTxRTPIP, nTxRTPPort, pCallId)  
{  
}
```

See Also

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

OnProvisionalResponse()

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

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

Syntax

```
void OnProvisionalResponse(  
    nLineNo,  
    nStatusCode,  
    pReasonPhrase  
)
```

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).

pReasonPhrase(string)

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

Example

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

See Also

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

OnFailureResponse()

The OnFailureResponse() event triggers when client dials a phone call and receives failure response from SIP server. The SIP request failure responses lie in the range of 4xx to 6XX (400 to 606). 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
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		
Global Failures 6xx			
600	Busy Everywhere	603	Decline
604	Does Not Exist Anywhere	606	Not Acceptable

Syntax

```
void OnFailureResponse(
    nLineNo,
    nStatusCode,
    pReasonPhrase
)
```

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).

pReasonPhrase(string)

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

Example

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

See Also

OnProvisionResponse(), OnRedirectResponse()

OnRedirectResponse()

The OnRedirectResponse() 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 OnRedirectResponse(  
    nLineNo,  
    nStatusCode,  
    pReasonPhrase,  
    pContact  
)
```

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 (300, 380 etc).

pReasonPhrase(string)

This parameter specifies SIP response reason phrase (Use Proxy, Moved Permanently etc).

pContact(string)

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

Example

```
OnRedirectResponse( nStatusCode, pReasonPhrase, pContact)  
{  
}
```

See Also

Disconnect(), OnProvisionResponse(), OnFailureResponse()

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()

OnFailToTransfer()

The OnFailToTransfer() event triggers when SIP server failed to acknowledges/accepts the call transfer request.

Syntax

```
void OnFailToTransferTransfer(  
                                nLineNo,  
                                nStatusCode,  
                                pReasonPhrase  
                                )
```

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

pReasonPhrase(string)

This parameter specifies SIP response reason phrase

Example

```
OnCallTransferAccepted(nLineNo, nStatusCode, pReasonPhrase)  
{  
}  
}
```

See Also

TransferCallEx()

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: OnProvisionalResponse()
Remote party/User accepts the call.
Event: OnConnected()
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: OnConnected()
Call/Voice session successfully establish.
Method:DisConnect()/OnDisconnectCall()
Remote Party/User hang up the phone.