SIP SOFTPHONE SDK

Android OS
TECHNICAL DOCUMENTATION
VERSION 2.2

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
TransferCall() 16
TransferCallEx()....17
HoldLine() 18
UnHoldLine() 19
IsLineConnected()....20
IsLineBusy()....21
IsLineHold() 23
OpenLine()....24
DigitDTMF()....26
DeselectAllVoiceCodec()....27
SelectAllVoiceCodec() 28
SelectVoiceCodec()....29
DeselectVoiceCodec() 30
EnableKeepAlive()....31
DisableKeepAlive() 32
SetLicenceKey()....33
GetVaxObjectError()....34
MuteLineSPK()....36
MuteLineMIC()....37
MicSetSoftBoost() 38
MicGetSoftBoost() 39
SpkSetSoftBoost()....40
SpkGetSoftBoost() 41
MicSetAutoGain()....42
MicGetAutoGain()....43
SpkSetAutoGain() 44
SpkGetAutoGain()....45
MuteMic()....46
MuteSpk()....47
VoiceChanger()....48
DonotDisturb()....49
RecordStart() 50
RecordStop()....51
RecordPause() 52
DiagnosticLog()....53
PostOneSecondTick()....54

PostSocketRecvSIP() 56
PostSocketRecvRTP() 58
PostMicDataPCM() 60

EXPORTED EVENTS 61
OnSuccessToRegister() 61
OnSuccessToReRegister 62
OnSuccessToUnRegister 63
OnTryingToRegister() 64
OnTryingToReRegister() 65
OnTryingToUnRegister() 66
OnFailToRegister() 67
OnFailToReRegister() 68
OnFailToUnRegister() 69
OnConnecting() 70
OnTryingToHold() 71
OnTryingToUnHold() 72
OnFailToHold() 73
OnFailToUnHold() 74
OnSuccessToHold() 75
OnSuccessToUnHold() 76
OnFailToConnect() 77
OnIncomingCall() 78
OnIncomingCallRingingStart() 79
OnIncomingCallRingingStop() 80
OnConnected() 81
OnProvisionalResponse() 82
OnFailureResponse() 83
OnRedirectResponse() 85
OnDisconnectCall() 86
OnCallTransferAccepted() 87
OnFailToTransfer() 88
OnIncomingDiagnostic() 89
OnOutgoingDiagnostic() 90
OnSocketOpenSIP() 91
OnSocketCloseSIP() 92
OnSocketSendSIP() 93
OnSocketOpenRTP() 94
OnSocketCloseRTP() 95
OnSocketSendRTP() 96
OnOpenMediaDevice() 97
OnCloseMediaDevice() 98
OnSpkDataPCM() 99

VAXVOIP BASED SOFTPHONE CALL FLOW 100
Execution Sequence of Methods/Events to Dial a Call 100
Execution Sequence of Methods/Events to Receive a Call 102

INTRODUCTION AND QUICK START

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

EXPORTED FUNCTIONS

InitializeEx()

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

Syntax

```
boolean  InitializeEx(
                    BindToListenIP,
                    ListenIP,
                    ListenPort,
                    UserName,
                    Login,
                    LoginPwd,
                    DisplayName,
                    DomainRealm,
                    SIPProxy,
                    SIPOutBoundProxy,
                    TotalLine
                    )
```

Parameters

BindToListenIP (boolean)

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

ListenIP (string)

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

ListenPort (integer)

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

UserName (string)

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

Login (string)

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

LoginPwd (string)
This parameter value specifies the password which is provided by IP-Telephony service provider or VoIP providers.

DisplayName (string)
This parameter value specifies the display name for user which is provided by IP-Telephony service provider or VoIP providers.

DomainRealm (string)
This parameter value is provided by IP-Telephony service provider or VoIP providers.

SIPProxy (string)
This parameter value is provided by IP-Telephony service provider or VoIP providers.

SIPOutBoundProxy (string)
This parameter value is provided by IP-Telephony service provider or VoIP providers.

NOTE: In some cases, ITSP (IP-Telephony service provider) supports outbound proxy. Outbound proxy is the only way to let the NAT/firewall user to make and receive phone calls.
If your service provider does not provide SIP outbound proxy then leave that field blank or ""

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

Return Value

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

Example

```
InitializeEx(False, "198.168.0.103", 5060, "8002", "8002", "1234", "abc",
            " ", "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(Expire)
```

Parameters

Expire (integer)
: The Expire parameter specifies the time interval (in seconds) 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(
                  LineNo,
                  DialNo,
                  InputDeviceId,
                  OutputDeviceId
                 )
```

Parameters

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

DialNo (string)
: This parameter specifies the user name or phone number to be dialed.

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

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

Return Value

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

Example

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

See Also

Connect(), Disconnect(), 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(
                LineNo,
                ToURI,
                InputDeviceId,
                OutputDeviceId
                )
```

Parameters

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

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

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

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

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(LineNo)
```

Parameters

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

Return Value

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

Example

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

See Also

DialCall(), Connect(), GetVaxObjectError()

AcceptCall()

The AcceptCall() function accepts the incoming call.

Syntax

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

Parameters

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

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

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

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

Return Value

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

Example

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

See Also

RejectCall(), GetVaxObjectError()

RejectCall()

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

Syntax

```
boolean RejectCall(CallId)
```

Parameters

CallId (string)
: The CallId 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(
                    LineNoA,
                    LineNoB
                    )
```

Parameters

LineNoA (integer)

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

LineNoB (integer)

This parameter value specifies the specific line. The LineNoB 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(), TransferCall(), GetVaxObjectError()

TransferCall()

The TransferCall() 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 TransferCall(
                        LineNo,
                        ToURI
                        )
```

Parameters

LineNo (integer)

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

ToURI (string)

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

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 = TransferCall(2, "SIP:8002@abc.com")
if(Result == 0) GetVaxObjectError()
```

See Also

AcceptCall(), GetVaxObjectError()

TransferCallEx()

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

Syntax

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

Parameters

LineNo (integer)

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

ToUserName (string)

This parameter specifies the 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() function puts a specific line on hold.

Syntax

```
boolean HoldLine(LineNo)
```

Parameters

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

Return Value

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

Example

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

See Also

UnHoldLine(), GetVaxObjectError()

UnHoldLine()

The UnHoldLine() function unholds a specific line.

Syntax

```
boolean UnHoldLine(LineNo)
```

Parameters

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

Return Value

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

Example

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

See Also

HoldLine(), GetVaxObjectError()

IsLineConnected()

The IsLineConnected() function get the status of line i.e. connected or free.

Syntax

```
boolean IsLineConnected(LineNo)
```

Parameters

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

Return Value

The function returns 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

Connect(), GetVaxObjectError()

IsLineBusy()

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

Syntax

```
boolean IsLineBusy(LineNo)
```

Parameters

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

Return Value

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

Example

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

See Also

OpenLine(), IsLineOpen()

IsLineOpen()

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

Syntax

```
boolean IsLineOpen(LineNo)
```

Parameters

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

Return Value

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

Example

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

See Also

OpenLine(), IsLineBusy()

IsLineHold()

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

Syntax

```
boolean IsLineHold(LineNo)
```

Parameters

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

Return Value

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

Example

```
Result = IsLineHold(3)
```

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(
                    LineNo,
                    BindToRTPRxIP,
                    RTPRxIP,
                    RTPRxPort
                    )
```

Parameters

LineNo (integer)

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

BindToRTPRxIP (boolean)

The BindToRTPRxIP 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.

RTPRxIP (string)

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

RTPRxPort (integer)

The RTPRxPort 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

IsLineOpen(), 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(
                    LineNo,
                     Digit
                     )
```

Parameters

LineNo (integer)

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

Digit (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 options for voice codecs.

Syntax

```
DeselectAllVoiceCodec()
```

Parameters

No parameters.

Return Value

No return value.

Example

```
DeselectAllVoiceCodec()
```

See Also

SelectAllVoiceCodec(), SelectVoiceCodec(), DeselectVoiceCodec()

SelectAllVoiceCodec()

The SelectAllVoiceCodec() function selects all options for voice codes.

Syntax

```
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 option for voice codec. Moreover the sequence of selection of voice codec decides the priority of codec i.e. the voice codec selected first has higher priority than the codec selected afterward.

Syntax

```
boolean SelectVoiceCodec(CodecNo)
```

Parameters

CodecNo (integer)

This parameter value ranges from 0-3 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(CodecNo)
```

Parameters

CodecNo (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(Seconds)
```

Parameters

Seconds (integer)
This parameter value specifies the time interval (in seconds) after which keep alive packets will be sent to keep the port open.

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

```
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

```
boolean SetLicenceKey(LicenceKey)
```

Parameters

LicenceKey (string)
The value of this parameter is license key provided by the company.

Return Value

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

Example

```
Result = SetLicenceKey("LicenseKey")
```

See Also

InitializeEx(), 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 function 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 input device, please re-try!
15	Unable to access output device, please re-try!
16	Microphone/Input device is not ready to use, please re-try!
17	Speaker/Output device is not ready to use, 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

InitializeEx() , SetLicenceKey()

MuteLineSPK()

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

Syntax

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

Parameters

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

Enable (boolean)
: The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to mute output voice stream otherwise zero.

Return Value

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

Example

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

See Also

MuteLineMIC(), GetVaxObjectError()

MuteLineMIC()

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

Syntax

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

Parameters

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

Enable (boolean)
: The Enable 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

MicSetSoftBoost()

The MicSetSoftBoost() method enables/disables microphone boost by increasing/decreasing the microphone sensitivity.

Syntax

```
boolean MicSetSoftBoost(Enable)
```

Parameters

Enable (boolean)
: The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to enable microphone boost otherwise zero 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

```
MicSetSoftBoost(1)
```

See Also

MicGetSoftBoost(), SpkSetSoftBoost(), SpkGetSoftBoost()

MicGetSoftBoost()

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

Syntax

```
boolean MicGetSoftBoost()
```

Parameters

No parameters.

Return Value

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

Example

```
MicGetSoftBoost()
```

See Also

MicSetSoftBoost(), SpkSetSoftBoost(), SpkGetSoftBoost()

SpkSetSoftBoost()

The SpkSetSoftBoost() method enables/disables volume boost of output voice stream by increasing/decreasing the speaker sensitivity.

Syntax

```
boolean SpkSetSoftBoost(Enable)
```

Parameters

Enable (boolean)
: The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to enable voice stream boost otherwise zero to disables 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

```
SpkSetSoftBoost(0)
```

See Also

MicGetSoftBoost(), MicSetSoftBoost(), SpkGetSoftBoost()

SpkGetSoftBoost()

The SpkGetSoftBoost() function gets the status of speaker boost i.e. enabled or disabled.

Syntax

```
boolean SpkGetSoftBoost()
```

Parameters

No parameters.

Return Value

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

Example

```
SpkGetSoftBoost()
```

See Also

MicSetSoftBoost(), SpkSetSoftBoost(), MicGetSoftBoost()

MicSetAutoGain()

The MicSetAutoGain() function sets the volume of input speech to a predetermined value irrespective of the user sound volume.

Syntax

```
boolean MicSetAutoGain(Value)
```

Parameters

Value (integer)
This parameter value specifies speech volume ranges between 0-20 (0 = Min Volume, 20 = Max Volume).

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

```
MicSetAutoGain(4)
```

See Also

MicGetAutoGain(), SpkSetAutoGain(), SpkGetAutoGain()

MicGetAutoGain()

The MicGetAutoGain() function gets the volume gain of the user input speech.

Syntax

```
integer MicGetAutoGain()
```

Parameters

No parameters.

Return Value

The function returns the volume gain set by MicSetAutoGain() method.

Example

```
MicGetAutoGain()
```

See Also

MicSetAutoGain(), SpkSetAutoGain(), SpkGetAutoGain()

SpkSetAutoGain()

The SpkSetAutoGain() function sets the volume of output voice stream to a predetermined value irrespective of the user sound volume

Syntax

```
boolean SpkSetAutoGain(Value)
```

Parameters

Value (integer)
This parameter value specifies speech volume ranges between 0-20 (0 = Min Volume, 20 = Max Volume).

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

```
SpkSetAutoGain(3)
```

See Also

MicGetAutoGain(), MicSetAutoGain(), SpkGetAutoGain()

SpkGetAutoGain()

The SpkGetAutoGain() function gets the volume gain of the output voice stream.

Syntax

```
integer SpkGetAutoGain()
```

Parameters

No parameters.

Return Value

The function returns the volume gain set by SpkSetAutoGain() method.

Example

```
SpkGetAutoGain()
```

See Also

MicGetAutoGain(), SpkSetAutoGain(), MicSetAutoGain()

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

```
MuteMic(Enable)
```

Parameters

Enable (boolean)
: The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to mute the microphone otherwise zero.

Return Value

No return value.

Example

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

See Also

MuteSpk(), GetVaxObjectError()

MuteSpk()

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

Syntax

```
MuteSpk(Enable)
```

Parameters

Enable (boolean)
The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to mute the speaker otherwise zero.

Return Value

No return value.

Example

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

See Also

MuteMic(), GetVaxObjectError()

VoiceChanger()

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

Syntax

```
boolean VoiceChanger(Pitch)
```

Parameters

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

Return Value

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

Example

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

See Also

DonotDisturb()

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

Syntax

```
boolean DonotDsiturb(Enable)
```

Parameters

Enable (boolean)
: The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to block ringing of 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(1)
```

See Also

RecordStart()

The RecordStart() function starts the recording of voice stream in a file.

Syntax

```
boolean RecordStart(FileName)
```

Parameters

FileName (string)
This parameter value specifies file for recording voice stream.

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

```
RecordStart("abc.wav")
```

See Also

GetVaxObjectError(), RecordStop()

RecordStop()

The RecordStop() function stops the recording of voice stream .

Syntax

```
RecordStop()
```

Parameters

No parameters.

Return Value

No return value.

Example

```
RecordStop()
```

See Also

RecordStart(), RecordPause()

RecordPause()

The RecordPause() method pauses the recording of voice stream on its current position.

Syntax

```
RecordPause(Enable)
```

Parameters

Enable (boolean)
: The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to pause recording otherwise zero.

Return Value

No return value.

Example

```
RecordPause(1)
```

See Also

RecordStart(), RecordStop()

DiagnosticLog()

The DiagnosticLog() function enables/disables the logging and monitoring of inbound/outbound SIP messages.

Syntax

```
boolean DiagnosticLog(Enable)
```

Parameters

Enable (boolean)

The Enable parameter value can be 0 or 1. Assign value 1 to this parameter to enable the logging and monitoring of inbound/outbound SIP messages otherwise zero to disable the logging and monitoring of inbound/outbound SIP messages.

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

```
DiagnosticLog(1)
```

See Also

OnIncomingDiagnostic(), OnOutgoingDaignostic(), GetVaxObjectError()

PostOneSecondTick()

The PostSecondTick() function sends tick to VaxVoIP component.

Android application initiates one second timer tick (repeatedly) and then use PostOneSecondTick() function to pass on that tick to VaxVoIP component and VaxVoIP component internally use that tick to perform certain task.

For further details, please see the sample code after downloading it from website.

Syntax

```
PostOneSecondTick()
```

Parameters

No parameters.

Return Value

No return value..

Example

```
TimerTask m_objTimerTask;

Timer m_objTimerTick;

final Handler m_objTickHandler = new Handler(Looper.getMainLooper());

class RunableTimerTick implements Runnable
{
	public void run()
	{
		m_objVaxVoIP.PostOneSecondTick();
	}
};


class TimerTaskEx extends TimerTask
{
	@Override
	public void run()
	{
		m_objTickHandler.post(new RunableTimerTick());
	}
};
```

```
public void StartTimer(int nMillisec)
{
        m_objTimerTask = new TimerTaskEx();

        m_objTimerTick = new Timer();
        m_objTimerTick.schedule(m_objTimerTask, nMillisec, nMillisec);
}
```

See Also

GetVaxObjectError()

PostSocketRecvSIP()

The PostSocketRecvSIP() function delivers the SIP packets to VaxVoIP component.

Android application implements communication socket and allows VaxVoIP component to send and receive SIP packets through this socket. Android application receives SIP packet via socket and then delivers it to VaxVoIP component by calling PostSocketRecvSIP().

For further details, please see the sample code after downloading it from website.

Syntax

```
PostSocketRecvSIP(
                    PacketData,
                    PacketSize,
                    FromIP,
                    FromPort
                    )
```

Parameters

PacketData (byte array)
: This parameter value specifies the SIP packet data.

PacketSize (integer)
: The PacketSize parameter value specifies the size of SIP packet.

FromIP (string)
: This parameter value specifies *From* IP address.

FromPort (integer)
: This parameter value specifies *From* port number.

Return Value

No return value.

Example

```
byte[] aData = new byte[MAX_UDP_DATAGRAM_LEN];
DatagramPacket objDgramPacket = new DatagramPacket(aData,
                                                   aData.length);
DatagramSocket.receive(objDgramPacket);

FromIP = objDgramPacket.getAddress().toString();
FromPort = objDgramPacket.getPort();

PostSocketRecvSIP(aData, objDgramPacket.getLength(), FromIP, FromPort)
```

See Also

PostSocketRecvRTP()

PostSocketRecvRTP()

The PostSocketRecvRTP() function sends the RTP(voice) data\packets to VaxVoIP component.

Communication sockets are implemented in Android application to allow VaxVoIP component to send/receive voice RTP packets. Android application receives voice RTP data and forward it to VaxVoIP component by calling PostSocketRecvRTP().

For further details, please see the sample code after downloading it from website.

Syntax

```
PostSocketRecvRTP(
                    LineNo
                    PacketData,
                    PacketSize,
                    FromIP,
                    FromPort
                    )
```

Parameters

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

PacketData (byte array)
: This parameter value specifies the RTP packet data.

PacketSize (integer)
: The PacketSize parameter value specifies the size of RTP packet.

FromIP (string)
: This parameter value specifies *From* IP address.

FromPort (integer)
: This parameter value specifies *From* port number.

Return Value

No return value.

Example

```
byte[] aData = new byte[MAX_UDP_DATAGRAM_LEN];
DatagramPacket objDgramPacket = new DatagramPacket(aData,
                                                   aData.length);
```

```
DatagramSocket.receive(objDgramPacket);
FromIP = objDgramPacket.getAddress().toString();


FromPort = objDgramPacket.getPort();

PostSocketRecvRTP(m_nLineNo, aData, objDgramPacket.getLength(),
                  FromIP, FromPort)
```

See Also

PostSocketRecvSIP()

PostMicDataPCM()

The PostMicDataPCM() function sends the captured microphone data to VaxVoIP component.

Android application is responsible for management of media devices as well as capturing and rendering of voice data. As voice data capturing is performed by Android application therefore by calling PostMicDataPCM() method, Android application forwards the microphone data to VaxVoIP component.

For further details, please see the sample code after downloading it from website.

Syntax

```
boolean PostMicDataPCM(
                        DataPCM,
                        SizePCM
                        )
```

Parameters

DataPCM (byte array)
: This parameter value specifies the microphone PCM data.

SizePCM (integer)
: The SizePCM parameter value specifies the size of PCM data.

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

```
int nReadSize = AudioRecord.read(aMicBuff, 0, aMicBuff.length);

PostMicDataPCM(aMicBuff, nReadSize);
```

See Also

OnSpkDataPCM(), OnOpenMediaDevice(), GetVaxObjectError()

EXPORTED EVENTS

OnSuccessToRegister()

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

Syntax

```
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

```
OnSuccessToReRegister()
```

Parameters

No parameters.

Example

```
OnSuccessToReRegister()
{
}
```

See Also

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

OnSuccessToUnRegister

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

Syntax

```
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

```
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

```
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 .

Syntax

```
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 been 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

```
OnFailToRegister(
                StatusCode,
                ReasonPhrase
                )
```

Parameters

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

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

Example

```
OnFailToRegister(StatusCode, ReasonPhrase)
{
}
```

See Also

OnFailToUnRegister(), OnFailToReRegister(), OnSuccessToRegister(), RegisterToProxy(), UnRegisterToProxy()

OnFailToReRegister()

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

Syntax

```
OnFailToReRegister(
                    StatusCode,
                    ReasonPhrase
                    )
```

Parameters

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

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

Example

```
OnFailToReRegister(StatusCode, ReasonPhrase)
{
}
```

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

```
OnFailToUnRegister(
                    StatusCode,
                    ReasonPhrase
                    )
```

Parameters

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

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

Example

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

See Also

0nSuccessToUnRegister(), OnSuccessToRegister(), OnTryingToUnRegister()
RegisterToProxy(), UnRegisterToProxy()

OnConnecting()

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

Syntax

```
OnConnecting(LineNo)
```

Parameters

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

Example

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

See Also

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

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

```
OnTryingToHold(LineNo)
```

Parameters

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

Example

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

See Also

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

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

```
OnTryingToUnHold(LineNo)
```

Parameters

LineNo (integer)

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

Example

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

See Also

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

OnFailToHold()

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

Syntax

```
OnFailToHold(LineNo)
```

Parameters

LineNo(integer)

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

Example

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

See Also

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

OnFailToUnHold()

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

Syntax

```
OnFailToUnHold(LineNo)
```

Parameters

LineNo (integer)

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

Example

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

See Also

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

OnSuccessToHold()

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

Syntax

```
OnSuccessToHold(LineNo)
```

Parameters

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

Example

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

See Also

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

OnSuccessToUnHold()

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

Syntax

```
OnSuccessToUnHold(LineNo)
```

Parameters

LineNo(integer)

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

Example

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

See Also

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

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

```
OnFailToConnect(LineNo)
```

Parameters

LineNo (integer)

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

Example

```
OnFailToConnect()
{
}
```

See Also

OnDisconnectCall(), Connect(), Disconnect()

OnIncomingCall()

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

Syntax

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

Parameters

CallId (string)
: The CallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

DisplayName (string)
: This Parameter value is provided by IP-Telephony service provider or VoIP providers.

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

FromURI (string)
: This parameter specifies *From* URI in incoming SIP call request.
SIP*:username@VaxVoIP-Listen-IP*
Username in *From* URI appears as caller-Id.

ToURI (string)
: This parameter specifies *To* URI in incoming SIP call request.
SIP*:username@domain/realm*
Username in ToURI appears as dial number.

Example

```
OnInComingCall(CallId, DisplayName, UserName, FromURI, ToURI)
{
}
```

See Also

AcceptCall(), RejectCall(), HoldLine()

OnIncomingCallRingingStart()

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

Syntax

```
OnIncomingCallRingingStart(CallId)
```

Parameters

CallId (string)
: The CallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

Example

```
OnIncomingCallRingingStart(CallId)
{
}
```

See Also

AcceptCall(), RejectCall(), OnIncomingCall(), OnIncomingCallRingingStop()

OnIncomingCallRingingStop()

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

Syntax

```
OnIncomingCallRingingStop(CallId)
```

Parameters

CallId (string)
: The CallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

Example

```
OnIncomingCallRingingStop(CallId)
{
}
```

See Also

AcceptCall(), RejectCall(), OnIncomingCall(), OnIncomingCallRingingStart()

OnConnected()

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

Syntax

```
onConnected(
        LineNo,
        TxRTPIP,
        TxRTPPort,
        CallId
        )
```

Parameters

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

TxRTPIP (string)
: This parameter value specifies the IP address of computer on which VaxVoIP receives voice streams.

TxRTPPort (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.

CallId (string)
: The CallId parameter value is a unique identifier for each incoming call. The value of this parameter is generated internally by the system.

Example

```
OnConnected(LineNo, TxRTPIP, TxRTPPort, CallId)
{
}
```

See Also

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

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

Parameters

LineNo (integer)

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

StatusCode (integer)

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

ReasonPhrase (string)

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

Example

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

See Also

DialCall(), Connect(), OnRedirectResponse(), OnFailureResponse()

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

```
OnFailureResponse(
                    LineNo,
                    StatusCode,
                    ReasonPhrase
                    )
```

Parameters

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

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

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

Example

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

See Also

OnProvisionalResponse(), 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

```
OnRedirectResponse(
                    LineNo,
                    StatusCode,
                    ReasonPhrase,
                    Contact
                    )
```

Parameters

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

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

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

Contact (string)
: This parameter value specifies the contact where SIP server will redirect the call.

Example

```
OnRedirectResponse(StatusCode, ReasonPhrase, Contact)
{
}
```

See Also

Disconnect(), OnProvisionalResponse(), OnFailureResponse()

OnDisconnectCall()

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

Syntax

```
OnDisconnectCall(LineNo)
```

Parameters

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

Example

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

See Also

OnFailToConnect(), Connect(), Disconnect()

OnCallTransferAccepted()

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

Syntax

```
OnCallTransferAccepted(LineNo)
```

Parameters

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

Example

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

See Also

TransferCallEx(), TransferCall()

OnFailToTransfer()

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

Syntax

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

Parameters

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

StatusCode (integer)
: This parameter specifies SIP response status code

ReasonPhrase (string)
: This parameter specifies SIP response reason phrase

Example

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

See Also

TransferCallEx()

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

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

Parameters

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

FromIP (string)
: This parameter value specifies *From* IP address.

FromPort (integer)
: This parameter specifies *From* port number.

Example

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

See Also

OnOutgoingDiagnostic()

OnOutgoingDiagnostic()

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

Syntax

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

Parameters

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

ToIP (string)
: This parameter value specifies the *To* IP address.

ToPort (integer)
: This parameter specifies the *To* port number.

Example

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

See Also

OnIncomingDiagnostic()

OnSocketOpenSIP()

The OnSocketOpenSIP() event triggers when VaxVoIP component sends socket open request to Android application for receiving/sending SIP packets.

Android application is responsible for opening/closing of communication sockets, so to send/receive data packets VaxVoIP component request Android application to open socket by triggering OnSocketOpenSIP() event.

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSocketOpenSIP(
                    ListenIP,
                    ListenPort
                    )
```

Parameters

ListenIP (string)

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

ListenPort (integer)

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

Example

```
OnSocketOpenSIP(ListenIP, ListenPort)
{
}
```

See Also

OnSocketCloseSIP(), PostSocketRecvSIP()

OnSocketCloseSIP()

The OnSocketCloseSIP() event triggers when VaxVoIP component sends socket close request to Android application.

As sockets are managed by Android application therefore whenever VaxVoIP component requires socket to be closed it send request to close the socket to Android application by triggering OnSocketCloseSIP() event.

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSocketCloseSIP()
```

Parameters

No parameters.

Example

```
OnSocketCloseSIP()
{
}
```

See Also

OnSocketOpenSIP(), PostSocketRecvSIP()

OnSocketSendSIP()

The OnSocketSendSIP() event triggers when VaxVoIP component sends SIP packet request to Android application.

VaxVoIP component sends SIP packet via socket, sockets are implemented in Android application, so to send SIP packet VaxVoIP component send request to Android application by triggering OnSocketSendSIP() event.

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSocketSendSIP(
                    PacketData,
                    PacketSize,
                    ToIP,
                    ToPort
                    )
```

Parameters

PacketData (byte array)
: This parameter value specifies the RTP packet data.

PacketSize (integer)
: The PacketSize parameter value specifies the size of RTP packet.

ToIP (string)
: This parameter value specifies the *To* IP address.

ToPort (integer)
: This parameter specifies the *To* port number.

Example

```
OnSocketSendSIP(PacketData, PacketSize, ToIP, ToPort)
{
}
```

See Also

PostSocketRecvSIP()

OnSocketOpenRTP()

The OnSocketOpenRTP() event triggers when VaxVoIP component sends socket open request to Android application to receive/send RTP packets.

Android application is responsible for opening/closing of communication sockets, so to send/receive voice RTP packets VaxVoIP component request Android application to open socket by triggering OnSocketOpenRTP() event.

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSocketOpenRTP(
                    LineNo
                    ListenIP,
                    ListenPort
                    )
```

Parameters

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

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

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

Example

```
OnSocketOpenRTP(LineNo, ListenIP, ListenPort)
{
}
```

See Also

OnSocketCloseRTP(), PostSocketRecvRTP()

OnSocketCloseRTP()

The OnSocketCloseRTP() event triggers when VaxVoIP component sends socket close request to Android application.

As sockets are managed by Android application therefore whenever VaxVoIP component requires socket to be closed it send request to close the socket to Android application by triggering OnSocketCloseRTP() event.

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSocketCloseRTP(LineNo)
```

Parameters

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

Example

```
OnSocketCloseRTP(LineNo)
{
}
```

See Also

OnSocketOpenRTP(), PostSocketRecvRTP()

OnSocketSendRTP()

The OnSocketSendRTP() event triggers when VaxVoIP component sends RTP packet request to Android application.

VaxVoIP component sends SIP packet via socket, sockets are implemented in Android application, so to send SIP packet VaxVoIP component send request to Android application by triggering OnSocketSendRTP() event.

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSocketSendRTP(
                    LineNo,
                    PacketData,
                    PacketSize,
                    ToIP,
                    ToPort
                    )
```

Parameters

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

PacketData (byte array)
: This parameter value specifies the RTP packet data.

PacketSize (integer)
: The PacketSize parameter value specifies the size of RTP packet.

ToIP (string)
: This parameter value specifies the *To* IP address.

ToPort (integer)
: This parameter specifies the *To* port number.

Example

```
OnSocketSendRTP(LineNo, PacketData, PacketSize, ToIP, ToPort)
{
}
```

See Also

PostSocketRecvRTP()

OnOpenMediaDevice()

The OnOpenMediaDevice() event triggers when VaxVoIP component sends media device open request to Android application.

Android application is responsible for management of media devices so media devices are opened/closed by Android application. To render or capture voice data the VaxVoIP component sends media device open request to Android application by calling OnOpenMediaDevice().

For further details, please see the sample code after downloading it from website.

Syntax

```
OnOpenMediaDevice(
                    LineNo,
                    DeviceMIC,
                    DeviceSPK
                    )
```

Parameters

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

DeviceMIC (integer)
: Reserve for future use.

DeviceSPK (integer)
: Reserve for future use.

Example

```
OnOpenMediaDevice(LineNo, DeviceMIC, DeviceSPK)
{
}
```

See Also

OnCloseMediaDevice()

OnCloseMediaDevice()

The OnCloseMediaDevice() event triggers when VaxVoIP component sends media device close request to Android application.

Capture/Release of media devices is implemented in Android application which allows VaxVoIP component to communicate with media devices for rendering/capturing of voice data. To close media device VaxVoIP component send media device close request to Android application by calling OnCloseMediaDevice().

For further details, please see the sample code after downloading it from website.

Syntax

```
OnCloseMediaDevice(LineNo)
```

Parameters

LineNo (integer)

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

Example

```
OnCloseMediaDevice(LineNo)
{
}
```

See Also

OnOpenMediaDevice()

OnSpkDataPCM()

The OnSpkDataPCM() event triggers when VaxVoIP component sends play PCM data request to Android application.

All operations related to media devices are managed/performed by Android application, so to render voice data VaxVoIP component send render PCM data request to Android application by calling OnSpkDataPCM() .

For further details, please see the sample code after downloading it from website.

Syntax

```
OnSpkDataPCM(
                DataPCM,
                SizePCM
                )
```

Parameters

DataPCM (byte array)
: This parameter value specifies the microphone PCM data.

SizePCM (integer)
: The SizePCM parameter value specifies the size of PCM data.

Example

```
OnSpkDataPCM(DataPCM, SizePCM)
{
}
```

See Also

PostMicDataPCM()

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

Socket implementation in Android application for SIP and RTP communication.
Method: InitializeEx()
Event: OnSocketOpenSIP()
SIP socket binds to listen port.
Method: OpenLine()
Event: OnSocketOpenRTP()
RTP socket binds to listen port.
Method: RegisterToProxy()
Event: OnSocketSendSIP()
Sends SIP packet using SIP socket.
Android application receives SIP packet on SIP socket.
Method: PostSocketRecvSIP()
Event: OnTryingToRegister()
Event: OnSuccessToRegister()
VaxVoIP component successfully registered with SIP server.
Method: DialCall()/Connect()
Event: OnOpenMediaDevice()

Android application captures the media device.
Event: OnConnecting()
Event: OnProvisionalResponse()
Remote party/User accepts the call.
Event: OnConnected()
Call/Voice session successfully establish.
Method/Event: Disconnect()/OnDisconnectCall()
Remote Party/User hang up the phone.
Event: OnCloseMediaDevice()
Android application releases the media device.

Execution Sequence of Methods/Events to Receive a Call

Socket implementation in Android application for SIP and RTP communication.
Method: InitializeEx()
Event: OnSocketOpenSIP()
SIP socket binds to listen port.
Method: OpenLine()
Event: OnSocketOpenRTP()
RTP socket binds to listen port.
Method: RegisterToProxy()
Event: OnSocketSendSIP()
Sends SIP packet using SIP socket.
Android application receives SIP packet on SIP socket.
Method: PostSocketRecvSIP()
Event: OnTryingToRegister()
Event: OnSuccessToRegister()
VaxVoIP component successfully registered with SIP server.
Event: OnIncomingCall()
Event: OnIncomingCallRingingStart()
Method: AcceptCall()
User receives the call.
Event: OnOpenMediaDevice()
Android application captures the media device.

Event: OnConnected()
Call/Voice session successfully establish.
Method/Event: Disconnect()/OnDisconnectCall()
Remote Party/User hang up the phone.
Event: OnCloseMediaDevice()
Android application releases the media device.