

METHOD DETAILS

This document specifies the Status and Error Code Numbers and also describes the events and functions supported by Vax Proxy Client SDK Components:

CODE NUMBERS:

- Error Codes
- Status Codes

EVENTS:

- OnStatusEvent
- OnTextEvent
- OnConnectionEvent
- OnAddConferenceEvent
- OnWhisperTextEvent
- OnServerTextEvent

METHODS:

- Initialize()
- UnInitialize()

- SetLicenceKey()
- GetVaxObjectError()

- EnableVoiceActivityDetection()
- DisableVoiceActivityDetection()

- EnableCryptography()
- DisableCryptography()

- RegisterToProxy()
- UnRegisterToProxy()

- Connect()
- Disconnect()

- AddInConf()
- AcceptAddInConf()
- CancelAddInConf()

- StartRecording()
- StopRecording()
- ResetRecording()
- SaveRecordingToWaveFile()

- SetSpkVolume()
- GetSpkVolume()
- SetMicVolume()
- GetMicVolume()

- EnableMicBoost()
- DisableMicBoost()
- IsMicBoostEnable()

- AcceptIncomingCall()
- CancelIncomingCall()

- SendTextMsg()
- SendWhisperTextMsg()
- SendTextToServer()

- DisableBusyState()
- EnableBusyState()

- DisableNotAvailableState()
- EnableNotAvailableState()

- MuteMic()
- MuteSpk()

- GetOutBoundDataRate()
- GetInBoundDataRate()
- GetOutBoundTotalBytes()
- GetInBoundTotalBytes()

ERROR CODES DETAIL:

ERROR CODES	DESCRIPTION
10	VAXOBJECT is not initialized properly, To initialize, the VaxVoice Object <i>Initialize</i> method should be called.
11	Cannot access the input device (Microphone) OR input device is already in use.
12	Cannot access the output device.
13	Cannot open local communication port, port is invalid or is already in use.
14	Fail to initialize voice compression manager.
15	Fail to initialize VaxVoice task manager.
16	Invalid unique id to Accept or Cancel the connection request.
17	Invalid License Key.

18	Connection to proxy server failed.
19	Unable to register to proxy server.
20	Connection is rejected by the proxy server.
21	You are not registered to the proxy server.
22	Connections limit exceeded on Proxy Server side.
23	Recording media initialization error or cannot open the recording tmp file.
24	Unable to open the wave file to save the recorded voice data.
25	Already in the process of adding a person into the conversation.
26	Uniqueld is already in the conference.
27	Provided Uniqueld is not in the conference.
28	Fail to open voice recording channel.
29	Connection is already connecting, this error occurs if you are already in the state of connecting and accepts an incoming connection.
30	Cryptography key is not provided while initializing the VaxVoice Object. Please see the <i>Initialize</i> method for further details.
31	Fail to access the Mic/Input device Volume OR Sound device does not support Mic volume feature.
32	Fail to access the Speaker/Output device Volume OR Sound device does not support Speaker volume feature.

33	You are blocked on proxy server side.
----	---------------------------------------

ERROR CODES	DESCRIPTION
50	Connection failed to connect to Remote Person.
51	Trying to connect to remote end.
52	Connection to the Remote end is Lost.
53	Connection Connected, you may start Conversation now, using headphones/speakers and microphone.
54	Remote end/person has closed the connection.
55	Remote end/person is busy. The status of the Person to which you are trying to connect is busy.
56	Remote end/person is unavailable. The status of the Person to which you are trying to connect is unavailable.
57	Remote end/person has rejected the connection request.
58	Connection to Proxy Server Lost.
59	Proxy server has closed the connection.
60	Person to which you want to add into the conversation is already having the conversation.
61	Trying to add into the conversation/conference.
62	Successfully added into the conversation / conference.

63	Person to which you want to add into the conversation/conference has cancelled the add conversation request.
64	Adding into the conversation/conference failed.
65	This status code notifies that Encrypted data is being received from the remote end and VaxVoice object is unable to decrypt it because Cryptography key is not provided while initializing the VaxVoice Object. Please see the <i>Initialize</i> method for further details.

EVENTS:

OnTextEvent

This Event is triggered by the VaxVoice control, when a BROADCASTED text message along with the any text data is received from the remote end.

Parameters Values:

- Text Message String
- Client UniqueId
- User Data (User data can be any text based information.)

OnWhisperTextEvent

This Event is triggered by the VaxVoice control, when a PRIVATE text message along with the text data (User Data) is received from the remote end.

Parameters Values:

- Text Message String
- Client UniqueId
- User Data

OnStatusEvent

This Event is triggered by the VaxVoice control, to notify about the connection status.

Parameters Values:

- Status Code Number
- Client UniqueId

OnConnectionEvent

This Event is triggered, to notify the listening end that client is requesting to start conversation.

Parameters Values:

- Client UniqueId

Remarks:

When a client tries to establish a connection to the listening end, this event is triggered with Client UniqueId value, to notify the listening end about the connection request. To accept the connection, *AcceptIncomingCall()* method can be used.

OnAddConferenceEvent

This Event is triggered, to notify the listening end that client is requesting to add into the conversation.

Parameters Values:

- Client UniqueId

Remarks:

When a client tries to add listening end into the conversation, this event is triggered with Client's UniqueId value, to notify the listening end about the add conversation request. To accept the connection, *AcceptAddInConf()* method can be used.

OnServerTextEvent

This Event is triggered by the VaxVoice control, to notify about the text message sent by the proxy Server.

Parameters Values:

- Text Message String

EXPORTED METHODS:

Initialize()

This method is called to initialize the VaxVoice control. It requires Port Number to listen for incoming connections and/or data. After initializing, VaxVoice control starts listening for incoming Connections.

It also requires the cryptography key or secret words to decrypt the incoming data and/or Encrypt the outgoing Voice/Text data. BLOW FISH cryptography is used for Encryption/Decryption.

Two methods *EnableCryptography* and *DisableCryptography* can be used to enable/disable the encryption on outbound Voice Stream and Text Messages. Please see the sample source code for more details.

Parameters:

- Listening Port Number
- CryptographyKey or Secret Words

Return Value:

Non-zero on success, otherwise 0, and a appropriate error code can be retrieved by calling *GetVaxObjectError()* method.

UnInitialize()

To uninitialized the VaxVoice control, this method can be used.

RegisterToProxy()

To register to Proxy Server, this method can be used.

Parameters:

- Your Unique Id
- Password
- User Data (Any other text information, which can be sent during the registration.)
- IP Address of Proxy Server
- Port of Proxy Server
- TimeOut

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

Remarks:

Unique Id is required to register to proxy server, that Unique Id can be your email address, login name, telephone number etc.

You can implement the Login & Password functionality very easily using this method.

Please see the sample source code for more details.

UnRegisterToProxy()

To close the connection to the Proxy Server, this method can be used.

SetLicenceKey()

Call this method, to set the License key.

Parameter:

- License Key Provided by VaxVoice

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

Remarks:

You must pay one-time License fee in order to get the License Key. After getting the License key, you will set it using this method and it will remove the evaluation message box & expiry.

GetVaxObjectError()

Call this method, to get the error for the last operation that failed.

Return Value:

- Error Code Number

EnableVoiceActivityDetection()

To enable the VAD (voice activity detection) feature. In this feature, VaxVoice starts sending voice digital data only when you speak.

Parameter:

- Mic Sensitivity, any value from 0 to 255.
- Silence Duration in seconds.

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

Remarks:

Normally, you can set value 170 for Mic Sensitivity and Silence Duration: 5 seconds.

DisableVoiceActivityDetection()

To disable the VAD (voice activity detection) feature.

EnableCryptography()

To enable the Encryption on outgoing Voice Stream and Text data. When this method is called, VaxVoice component starts encrypting the outbound Voice Stream and Text messages before sending it to the Remote end over the internet.

Remote end receives the encrypted Voice Stream/Text messages and decrypt it using the provided Cryptography key. So on both ends the Cryptography keys must be the same.

BLOW FISH cryptography is used for Encryption/Decryption.

DisableCryptography()

To disable the Cryptography feature on outgoing Voice Stream and Text data. When this method is called, VaxVoice component stops encrypting the Voice Stream and Text messages and starts sending the plain data to the remote end over the Internet.

Connect()

This method is called to send connection request to listening end.

Parameters:

- To UniqueId
- Timeout to connect in seconds

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

Remarks:

This method tries to establish the connection to the remote listening end and the Status Event Codes triggers accordingly.

You can also send any text data along with the connection request and remote end will receive that text data into *OnConnectionEvent* event.

Disconnect()

Calling this method closes the voice session.

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

AddInConf()

To add more persons into the conversation, this method can be called.

Parameters:

- To UniqueId
- Timeout in seconds

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

Remarks:

This method tries to the remote listening end into the conversation and the Status Event Codes triggers accordingly.

You can also send any text data along with the add conference request and remote end will receive that text data into *OnAddConferenceEvent* event.

AcceptAddInConf()

This method is used to accept the incoming add conversation request

Parameter:

- Client UniqueId
- Timeout to connect in seconds

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

CancelAddInConf()

This method is used to cancel the incoming add conversation request.

Parameter:

- Client UniqueId

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

StartRecording()

This method is used to start the conversation recording.

Remarks:

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

StopRecording()

This method is used to stop the conversation recording.

Remarks:

Call to this method stops storing data into recording tmp file.

ResetRecording()

This method can be call to reset/clear the recording buffer.

Remarks:

Call to this method clears all the saved digital data from the recording tmp file.

SaveRecordingToWaveFile()

To save the recording tmp file voice data into wave (.wav) file this method can be use.

Parameter:

- File name (.wav)

Remarks:

Call to this method save the tmp voice data into wave (.wav) file.

EnableBusyState()

Call to this method enables the busy state.

Remarks:

Call to this method changes the current person status to busy. Suppose two people (A & B) are having voice conversation. During the conversation Person B sets his status to busy and then if another Remote Person C, tries to connect to Person B, Person C will get the STATUS CODE: 55, which represents the Person you are connecting to, is Busy.

DisableBusyState()

Call to this method, disables the busy state.

EnableNotAvailableState()

Call to this method enables the Not-Available state.

Remarks:

Call to this method, change the current Person's state to UN-AVAILABLE. For example, Person A is listening on a specific port and call to this method will change his status to unavailable, if any Remote Person B connects for voice conversation, the connection will not be established and the Remote Person B will get the STATUS CODE: 56

DisableNotAvailableState()

Call to this method, changes the state from UN-AVAILABLE to AVAILABLE.

MuteSpk()

To mute the speaker, this method can be called. Muting the speaker does not affect the Master Mute Control.

Parameter:

- Boolean Value 0 or 1

Return Value:

Non-Zero on success, otherwise 0

MuteMic()

To mute the microphone, this method can be called. Muting the Microphone does not affect the Master Mute Control. It simply blocks the outbound voice streaming data.

Parameter:

- Boolean Value 0 or 1

Return Value:

Non-Zero on success, otherwise 0

EnableMicBoost()

Call the method to increase the Microphone sensitivity.

DisableMicBoost()

To disable the Mic Boost.

IsMicBoostEnable()

Call the method to get Mic Boost is enabled or disabled.

GetSpkVolume()

Call to this method returns the speaker volume value between Range [0 to 255], where

0 = Min Volume

255 = Max Volume

Return Value:

Speaker Volume value on success, otherwise -1 and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

SetSpkVolume()

To set the Output volume, this method can be called. The value range should be between 0 to 255.

Parameter:

- Volume Value between [0 – 255] Range

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

GetMicVolume()

Call the method to get the Microphone Volume. Volume value is returned, in the range of [0 to 255], where

0 = Min Volume

255 = Max Volume

Return Value:

Microphone Volume value on success, otherwise -1 and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

SetMicVolume()

Call the method to set the Microphone Volume, the value range should between 0 to 255, where

0 = Min Volume

255 = Max Volume

Parameter:

- Volume Value between [0 – 255] Range

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

AcceptIncomingCall()

This method is used to accept the incoming connection.

Parameter:

- Client UniqueId
- Timeout to connect in seconds

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

CancelIncomingCall()

This method is used to ignore the incoming connection.

Parameter:

- Client UniqueId

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

SendTextMsg()

To broadcast text message along with any text data to all conference members. You can also send any text information along with the text message and remote end will receive that text data into *OnTextEvent* event.

Parameter:

- Text Message to broadcast
- Any text data

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

SendWhisperTextMsg()

To send text message to a specific conference member. You can also send any text data along with the text message and remote end will receive that text data into *OnWhisperTextEvent* event.

Parameter:

- Text Message to Send
- Client UniqueId
- Any text data

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

SendTextToServer()

To send text message to the proxy server. You can also send any text data along with the text message and proxy server will receive that text data into server side *OnClientTextEvent* event.

Parameter:

- Text Message to Send
- Any text data
- Server IP
- Server Port

Return Value:

Non-zero on success, otherwise 0, and a specific error code can be retrieved by calling *GetVaxObjectError()* method.

GetOutBoundTotalBytes()

Return Value:

Total Outbound bytes, since the VaxVoice Object is initialized.

GetInBoundTotalBytes()

Return Value:

Total Inbound bytes, since the VaxVoice object is initialized.

GetOutBoundDataRate()

Return Value:

Outbound Bytes per second.

GetInBoundDataRate()

Return Value:

Inbound Bytes per second.