

Agora Windows How-to Guide

Contents

Demo App.....	1
Requirements:	1
Demonstration.....	1
Sample Program.....	5
Create AgoraAudio object.....	5
Join Call	5
Mute Call.....	5
Leave Call.....	5
AudioEventHandler Interface	5

This manual explains how to use the Agora voice SDK by presenting a demo app. The demo app is included when you unzip the SDK in the sample folder.

Demo App

The demo app shows the basics of how to join a call and leave a call.

Requirements:

- Microsoft Visual C++ 2008 or greater
- Windows Mobile Professional Developers tool kit
- Windows mobile phone. Or you can use the emulator. If you have two phones that is best so that you can make a phone call from one device to another.
- Add the AgoraAudioSDK\include directory to the INCLUDE directories of your project.
- Add the 'AgoraAudioSDK\lib' directory to the LIB directories of your project and make sure mediasdk.lib is linked with your project.
- Copy the dlls under AgoraAudioSDK\dll to the directory where your executable file is located.

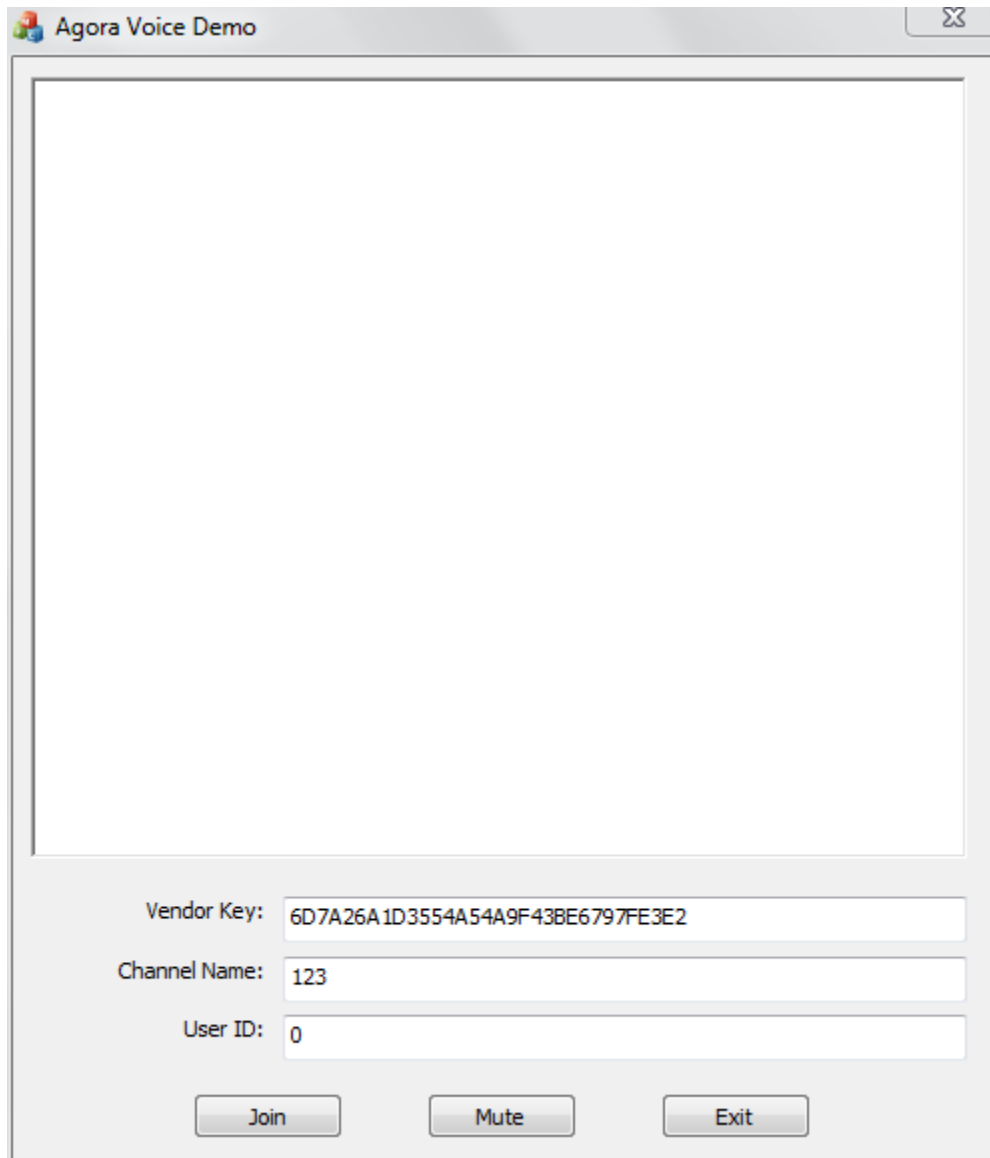
Demonstration

There are four basic functions to the Agora Voice SDK:

1. Initialization
2. Join Channel
3. Leave Channel

4. Mute Call

We explain those below with code samples. But first run the demo app to see what it does.

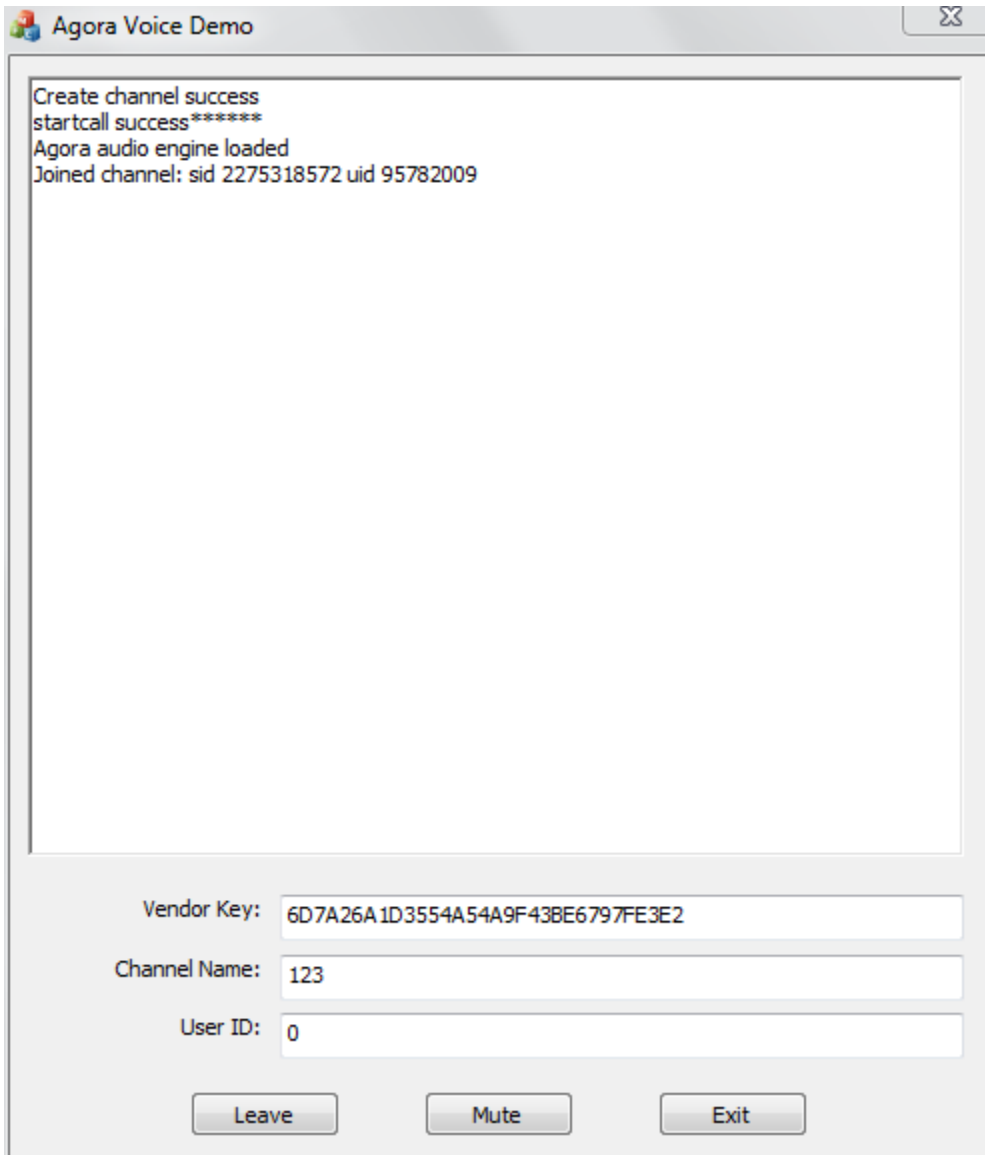


Enter the three fields explained below then press *Join* to join the call on the desired channel. You can do this even if no one else is in the channel, meaning you are the first person to join the call.

Vendor Key—this is the license key supplied to the Agora Voice customer to make calls over the Agora Voice cloud.

Channel Name—this is the name of the channel you can join. In the example below we use “1,” but it could be something like “conference call” or “game XYZ.”

User ID—if you leave this blank, the AgoraVoice object will create an ID for you. This uniquely identifies the parties to the call. For this demo leave it blank.



The screenshot shows a window titled "Agora Voice Demo". Inside, there is a text area with the following log messages:

```
Create channel success
startcall success*****
Agora audio engine loaded
Joined channel: sid 2275318572 uid 95782009
```

Below the text area are three input fields:

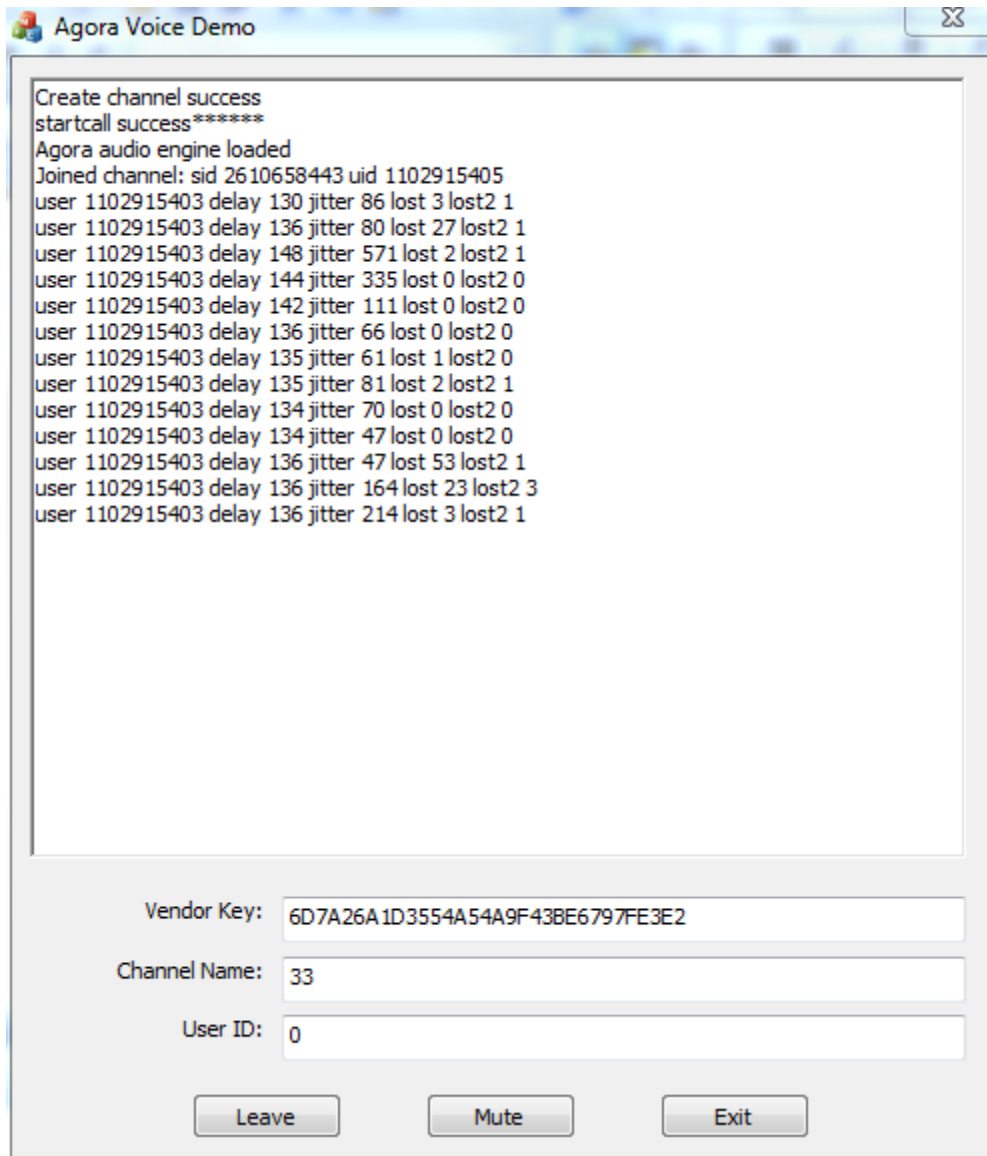
- Vendor Key: 6D7A26A1D3554A54A9F43BE6797FE3E2
- Channel Name: 123
- User ID: 0

At the bottom of the window are three buttons: "Leave", "Mute", and "Exit".

Channel joined successfully.

sid—this is integer representation of the channel number your entered, which in this case is “123.”

uid—User id. Since you entered “0” it created the user id shown above.



Now that you have joined the channel, the screen updates with call quality information. If a second person joins you will see information from that caller as well.

delay—voice delay in ms.

Jitter—VoIP jitter occurs when the data packets sent for your call are sent and received with significant variations in their timing.

lost—lost ratio in percentage, from 0 to 99.

Sample Program

The four basic operations initialize, join, leave, and mute are called as below:

Create AgoraAudio object

```
AgoraAudio(IAgoraAudioEventHandler* handler)
```

See information below on the AgoraAudioEventHandler.

Join Call

```
const char* key = "key granted by Agora";
const char* channel = "channel to join";
const char* extraInfo = "extra info you pass to SDK";
unsigned int uid = (put 0 to have Agora create a user id)
pAgoraAudio->joinChannel(key, channel, extraInfo, uid);
```

Mute Call

```
void mute(bool mute);
```

Leave Call

```
pAgoraAudio->leave();
```

AudioEventHandler Interface

The AgoraAudioEventHandler has these virtual functions that you can implement. onQuality and onJoinSuccess are callback methods to give you information on call quality. See the SDK document for a description of each of these functions and parameters.

```
virtual void onLoadAudioEngineSuccess() {}
virtual void onGetAudioSvrAddrSuccess(const char* msg) {}
virtual void onJoinSuccess(const char* msg) {}
virtual void onError(int rescode, const char* msg) {}
virtual void onLogEvent(const char* msg) {}
virtual void onQuality(unsigned int uid, unsigned short rtt, unsigned
short lost, unsigned short jitter) {}
```