

# Agora Audio SDK Introduction - Windows v0.5.1

- © 2014 AgoraLab, Inc. All rights reserved.
- Drafted @22/Jul/2014
- [support@agoralab.co](mailto:support@agoralab.co)

## 1. Intergrate Agora Audio SDK to your project

- Agora Audio SDK requires Visual C++ 2008 x86 runtime libraries on a computer that does not have Visual C++ 2008 installed.
- 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 dlls under `AgoraAudioSDK\dll` to the directory where your executable file places.

## 2. Class `AgoraAudio`

The APIs listed below are quite self explanatory. Methods exposed in the class `AgoraAudioKit` are asynchronous (except `joinChannel`), you can call them in the UI threads.

### 2.1 Initialization

```
- AgoraAudio(IAgoraAudioEventHandler* hanlder)
```

#### Description:

- This is the constructor for `AgoraAudio` Class. It will return an object of `AgoraAudio` class
- The methods accept an object of `IAgoraAudioEventHandler` class as the only argument. On error or on receiving response for the API request, the respective callback methods from the `IAgoraAudioEventHandler` object are invoked.

### 2.2 Join Channel

```
- void joinChannel(const char* vendorKey,  
                  const char* channelName,  
                  const char* info,  
                  unsigned int uid)
```

#### Description:

- This method requests the Agora Audio cloud service to create a channel specified by the channel name. clients specified with the same channel name can speak and listen to each other. Please Note that the method may block the calling thread for a moment since internally it calls gethostbyname to retrieve a list of access point server addresses. It is NOT recommended to call this method in the main UI thread.
- **vendorKey**: Account level credentials issued by AgoraLab company to each vendor.
- **channelName**: Joining in the same **channelName** indicates those clients have entered in one room referred above.
- **info**: this optional argument can be whatever the programmer likes personally
- **uid**: this optional argument is the unique ID for each member in one channel. Make sure to get rid of the issue of "duplicated ID in one channel", if SDK user wants to fill in this argument. Uid management will be taken over if it is 0 by default.

## 2.3 Leave Channel

```
- void leave()
```

#### Description:

- Leave the channel you joined before. after leaving a channel, all related resources are freed.

## 2.4 Mute

```
- void mute:(bool bMute);
```

#### Description:

- Mute the client itself by calling `mute(true)`, unmute it by calling `mute(false)`.

### 3. Interface **IAgoraAudioEventHandler**

```
class IAgoraAudioEventHandler
{
public:
    virtual void onLoadAudioEngineSuccess() = 0;

    virtual void onGetAudioSvrAddrSuccess(const char* msg) = 0;

    virtual void onJoinSuccess(const char* msg) = 0;

    virtual void onError(int rescode, const char* msg) = 0;

    virtual void onLogEvent(const char* msg) = 0;

    virtual void onQuality(unsigned int uid, unsigned short rtt, unsigned
short lost, unsigned short jitter) = 0;

};
```

The IAgoraAudioEventHandler interface is provided for your application to receive events from the SDK. Don't call blocking functions within the callbacks.

### 4. Example

To creat an AgoraAudio object:

```

class AgoraAudioEventHandler: public IAgoraAudioEventHandler
{
public:
    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) {}

};

static AgoraAudioEventHandler eh;

AgoraAudio *pAgoraAudio = new AgoraAudio(&eh);

```

To join a 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 = UNIQUE_IDENTIFICATION_FOR_EACH_USER; // e.g. 321 or just
0

pAgoraAudio->joinChannel(key, channel, extraInfo, uid);

```

To leave a call:

```

pAgoraAudio->leave();

```

To mute/unmute:

```
pAgoraAudio->mute(true); // mute
```

```
pAgoraAudio->mute(false); // unmute
```