Run Voice Chat within Your Unity Application

This document shows how you can configure and run your Unity Application with Agora IO's Voice SDK. The SDK supports iOS, Android, MacOS and Windows platform on Unity. Note that the functionalities of the Voice SDK is a subset of the Video SDK.

Prerequisites

- Unity Editor (2017 LTS or above)
- A developer account with Agora.io

Getting Started

Although you may start the integration with your existing project, in this short tutorial we will just use the demo project included in the SDK.

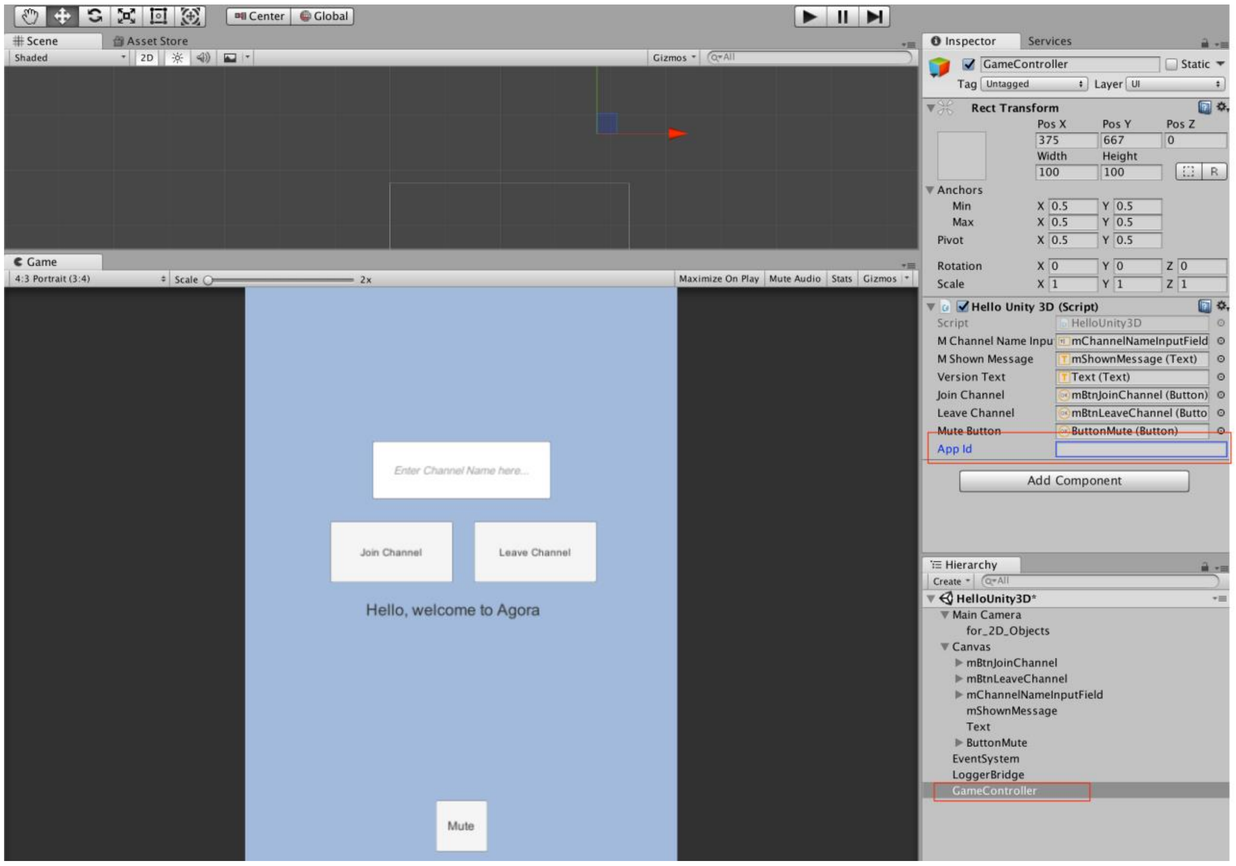
Open a new Unity project, and navigate to Unity Asset Store and search for "Agora Voice SDK". Download and import the assets.

Add Your AppID

Before you can build and run the project, you will need to add your AppID to the configuration. Go to your developer account's project console, create a new AppId or copy the AppID from an existing project. Perform the following steps:

1. Open the HelloUnity3D scene.
2. Select the `GameController` from the Unity Editor's Hierarchy panel.
3. The `GameController` game object has a property `App ID`, this is where you will add your Agora App ID.

See the following screen capture:



GameController Static
Tag Untagged Layer UI

Rect Transform

	Pos X 375	Pos Y 667	Pos Z 0
	Width 100	Height 100	
Anchors Min	X 0.5	Y 0.5	
Max	X 0.5	Y 0.5	
Pivot	X 0.5	Y 0.5	
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Hello Unity 3D (Script)

Script	HelloUnity3D
M Channel Name Inpu	mChannelNameInputField
M Shown Message	mShownMessage (Text)
Version Text	Text (Text)
Join Channel	mBtnJoinChannel (Button)
Leave Channel	mBtnLeaveChannel (Butto
Mute Button	ButtonMute (Button)
App Id	

At this step, you should be able to test the demo App within the Unity Editor. Just input your selected channel name to the input field and click Join. However, since this is SDK is built for interactive communication, you will need a separate device to run the same application in order to test voice chat.

Player Settings for Building the Sample Application

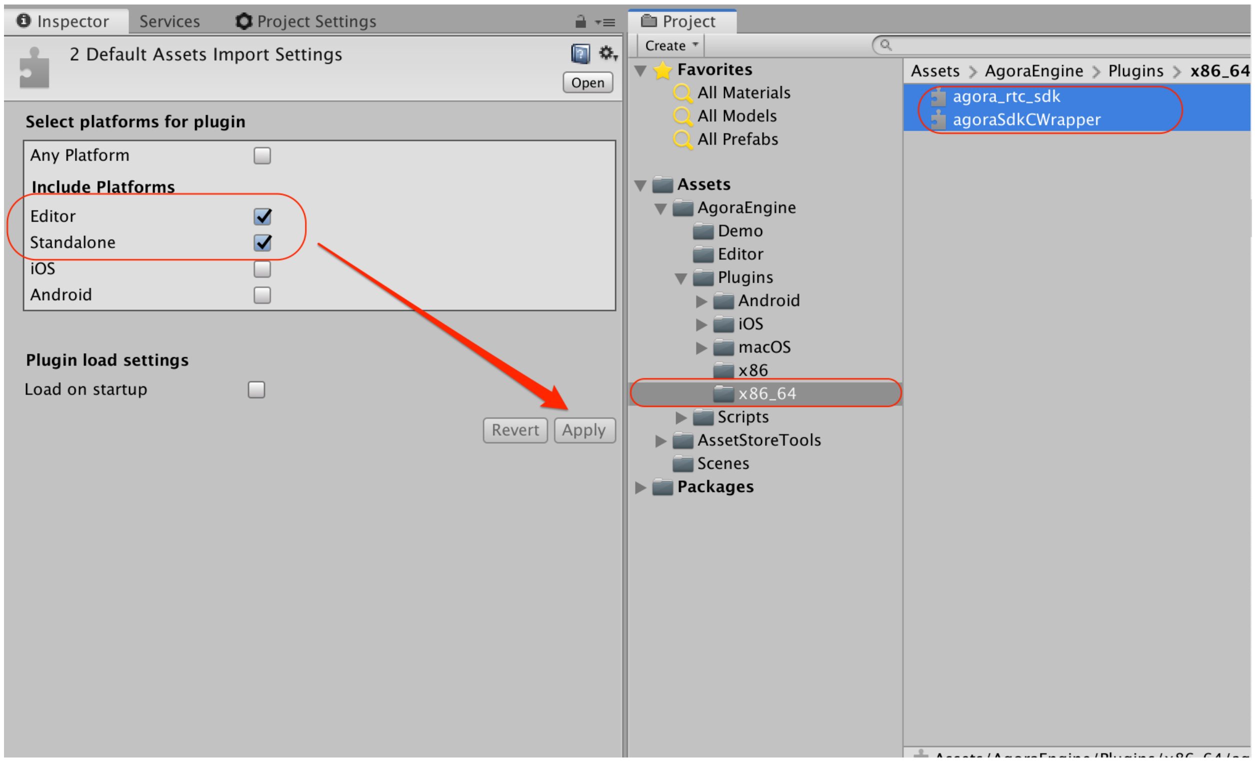
We will discuss the configuration for the four supported platforms in this section.

Common Setting

Open the **Build Settings** and drag HelloUnity3D from Demo folder into the “Scenes in Build” list.

Setting Plugin Identities

For **Windows** builds, go into Assets->Plugins->x86_64 folder, select “Editor” and “Standalone” and then click Apply.



Select platforms for plugin

Any Platform ☐

Include Platforms

Editor ✓

Standalone ✓

iOS ☐

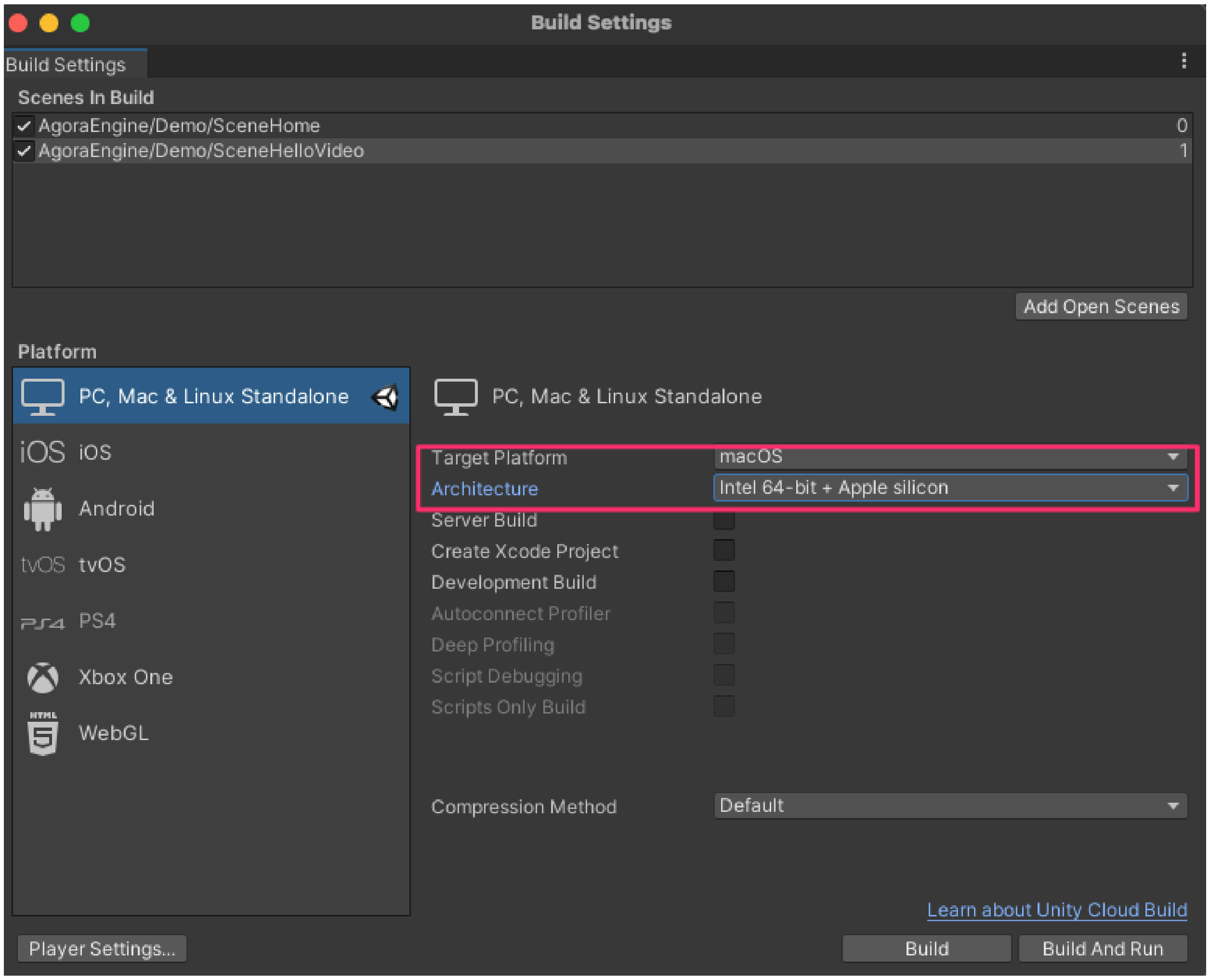
Android ☐

Plugin load settings

Load on startup ☐

To ensure that the build plugin libraries don’t collide, disable the identities of the files in the x86 folder.

For MacOS builds on Unity 2020.2 and up, make sure build target CPU type is selected to match library identities:



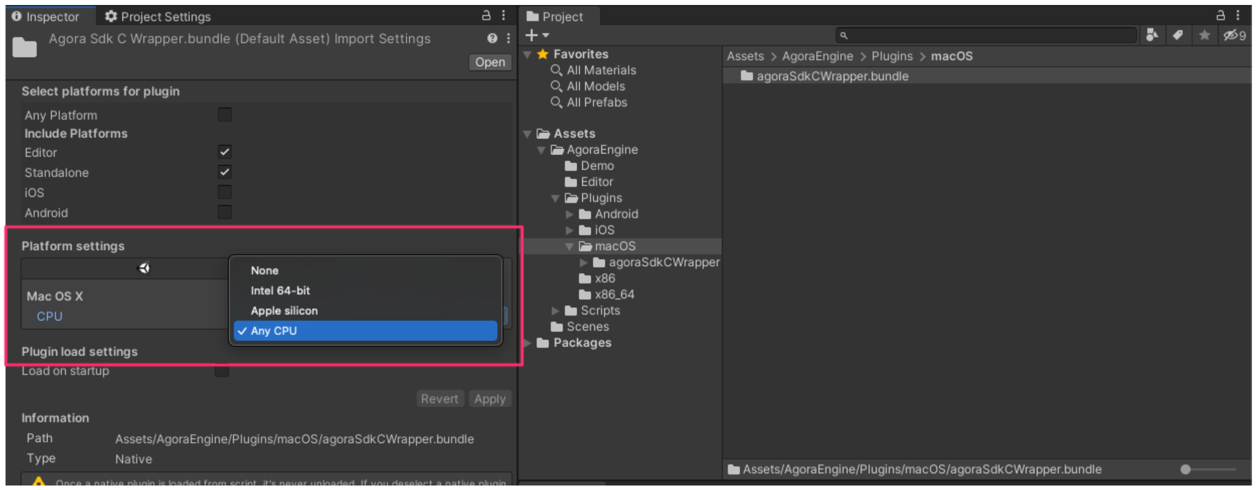
Build Settings

Scenes In Build

- ✓ AgoraEngine/Demo/SceneHome 0
- ✓ AgoraEngine/Demo/SceneHelloVideo 1

Platform: PC, Mac & Linux Standalone

Target Platform	macOS
Architecture	Intel 64-bit + Apple silicon
Server Build	☐
Create Xcode Project	☐
Development Build	☐
Autoconnect Profiler	☐
Deep Profiling	☐
Script Debugging	☐
Scripts Only Build	☐
Compression Method	Default



Agora Sdk C Wrapper.bundle (Default Asset) Import Settings

Select platforms for plugin

Any Platform	☐
Include Platforms	
Editor	✓
Standalone	✓
iOS	☐
Android	☐

Platform settings

Mac OS X — CPU: None / Intel 64-bit / Apple silicon / [✓ Any CPU]

Plugin load settings

Load on startup ☐

Information

Path	Assets/AgoraEngine/Plugins/macOS/agoraSdkCWrapper.bundle
Type	Native

Once a native plugin is loaded from script, it's never unloaded. If you deselect a native plugin

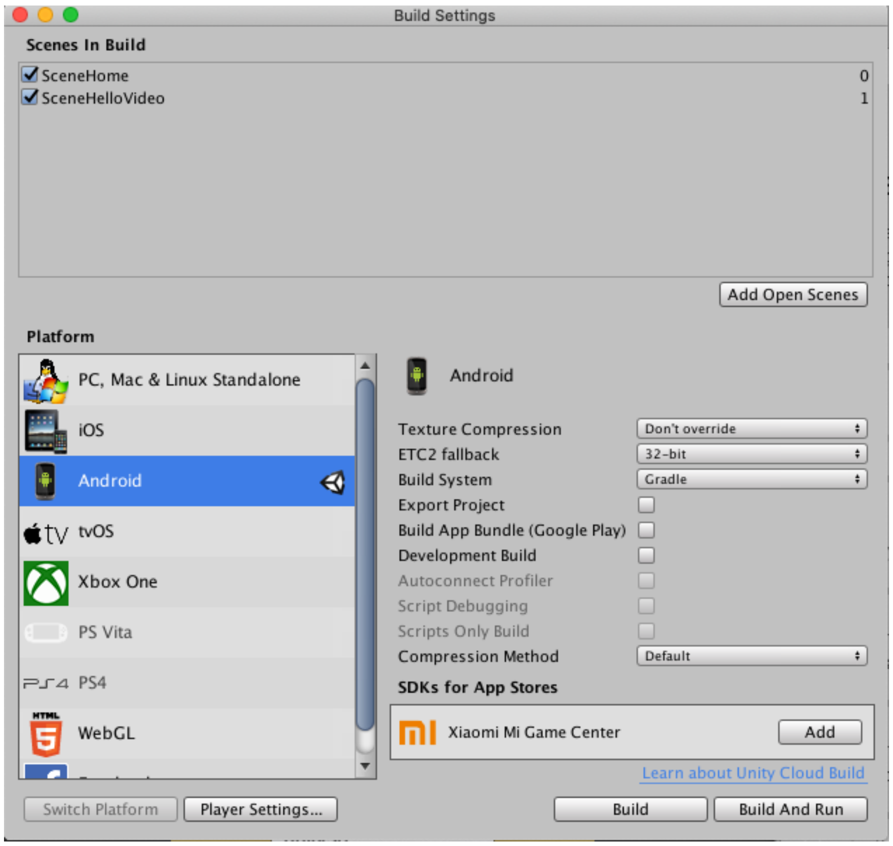
Assets > AgoraEngine > Plugins > macOS

agoraSdkCWrapper.bundle

Assets/AgoraEngine/Plugins/macOS/agoraSdkCWrapper.bundle

Android Build

Select **Android** from the platform list and click Switch Platform.



Build Settings

Scenes In Build

☑ SceneHome 0
☑ SceneHelloVideo 1

Add Open Scenes

Platform

- PC, Mac & Linux Standalone
- iOS
- Android
- tvOS
- Xbox One
- PS Vita
- PS4
- WebGL

Android

Texture Compression	Don't override
ETC2 fallback	32-bit
Build System	Gradle
Export Project	☐
Build App Bundle (Google Play)	☐
Development Build	☐
Autoconnect Profiler	☐
Script Debugging	☐
Scripts Only Build	☐
Compression Method	Default

SDKs for App Stores

Xiaomi Mi Game Center — Add

Learn about Unity Cloud Build

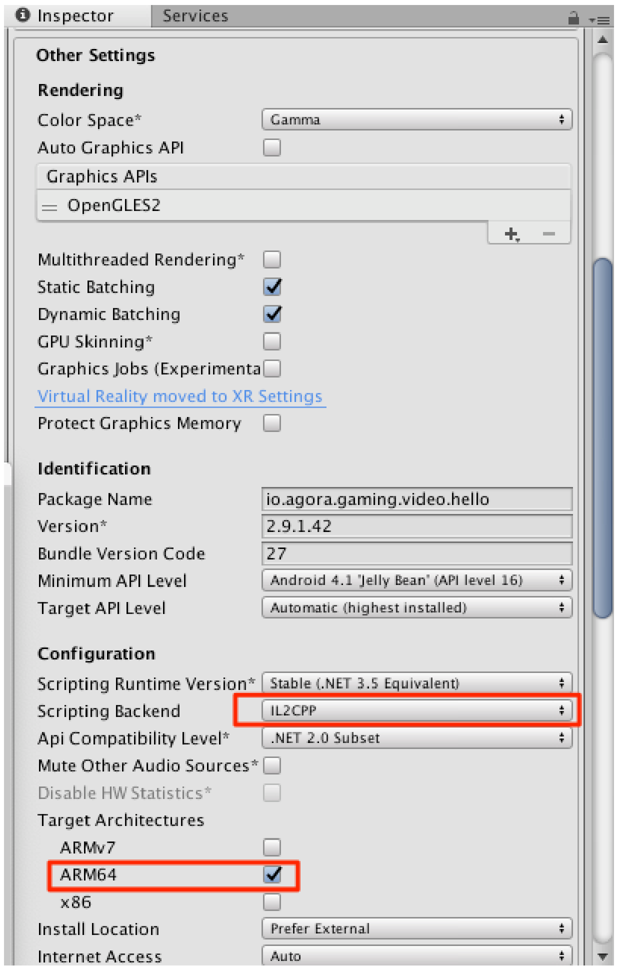
Switch Platform | Player Settings... | Build | Build And Run

Android Build

Once Unity finishes the setup process, open the Player Settings and set a unique package name, e.g., io.agora.gaming.hello.

In order to comply with Google's 64 bit App requirement, make sure the following setting is selected:

1. Change the Scripting backend to IL2CPP.
2. Select ARM64 for the Target Architecture.



Other Settings

Rendering

Color Space* Gamma

Auto Graphics API ☐

Graphics APIs

OpenGLES2

Multithreaded Rendering* ☐

Static Batching ✓

Dynamic Batching ✓

GPU Skinning* ☐

Graphics Jobs (Experimenta ☐

Virtual Reality moved to XR Settings

Protect Graphics Memory ☐

Identification

Package Name io.agora.gaming.video.hello

Version* 2.9.1.42

Bundle Version Code 27

Minimum API Level Android 4.1 'Jelly Bean' (API level 16)

Target API Level Automatic (highest installed)

Configuration

Scripting Runtime Version* Stable (.NET 3.5 Equivalent)

Scripting Backend IL2CPP

Api Compatibility Level* .NET 2.0 Subset

Mute Other Audio Sources* ☐

Disable HW Statistics* ☐

Target Architectures

ARMv7 ☐

ARM64 ✓

x86 ☐

Install Location Prefer External

Internet Access Auto

Android Build Settings

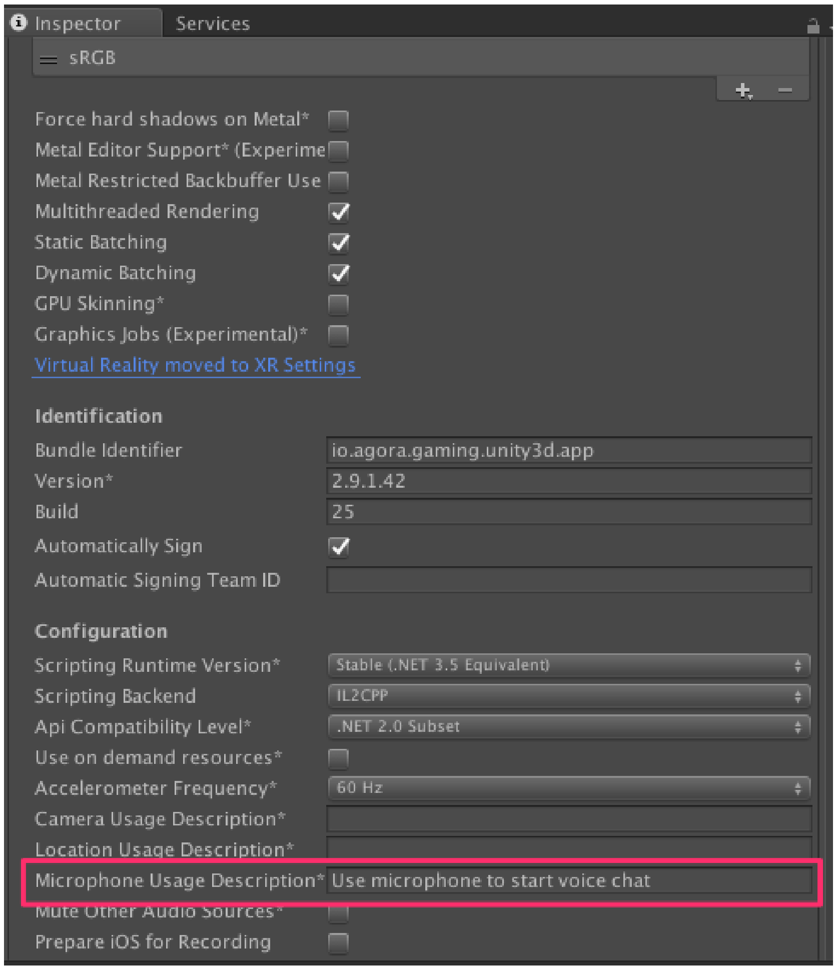
Leave other settings as is. For AR/VR enabled Applications, please refer to the separate README file.

iOS Build

Select **iOS** from the platform list and click Switch Platform.

The default build setting should work for most cases. The only custom settings are:

1. Change the Bundle Identifier to your own Bundle identifier so XCode can properly codesign the application.
2. Ensure the microphone permission has a description to allow the user to know why the microphone is being accessed by the application



Force hard shadows on Metal*	☐
Metal Editor Support* (Experime	☐
Metal Restricted Backbuffer Use	☐
Multithreaded Rendering	✓
Static Batching	✓
Dynamic Batching	✓
GPU Skinning*	☐
Graphics Jobs (Experimental)*	☐

Identification

Bundle Identifier	io.agora.gaming.unity3d.app
Version*	2.9.1.42
Build	25
Automatically Sign	✓
Automatic Signing Team ID	

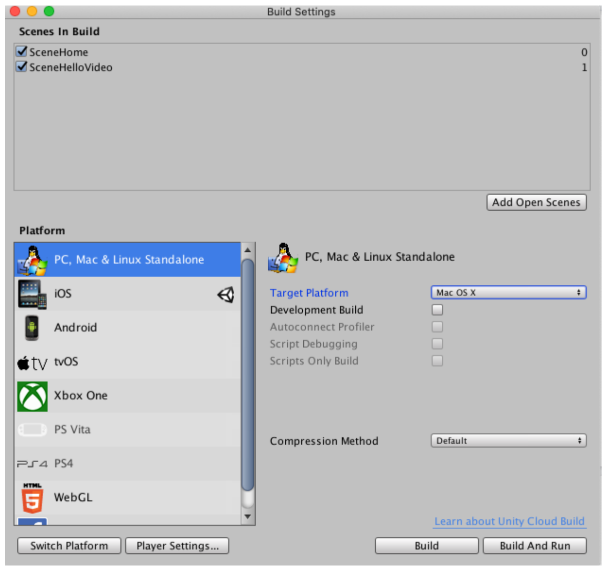
Configuration

Scripting Runtime Version*	Stable (.NET 3.5 Equivalent)
Scripting Backend	IL2CPP
Api Compatibility Level*	.NET 2.0 Subset
Use on demand resources*	☐
Accelerometer Frequency*	60 Hz
Camera Usage Description*	
Location Usage Description*	
Microphone Usage Description*	Use microphone to start voice chat
Mute Other Audio Sources*	☐
Prepare iOS for Recording	☐

iOS Build Settings

MacOS Build

Select **“PC, Mac & Linux Standalone”** from the platform list and click Switch Platform. Then Select “**Mac OS X**” for the Target Platform. Fill in the appropriate Mac App Store Options are needed.

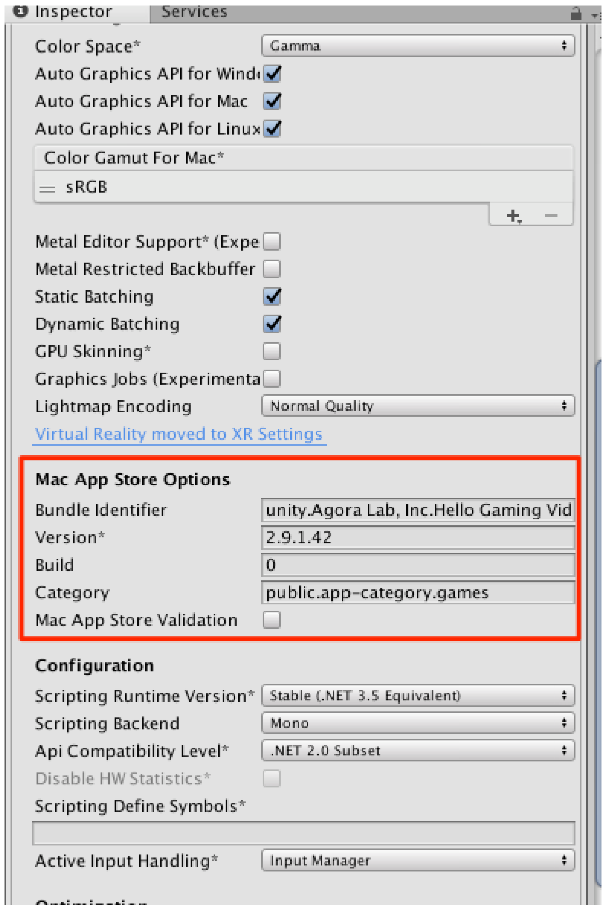

Build Settings

Scenes In Build

- ✓ SceneHome 0
- ✓ SceneHelloVideo 1

Platform: PC, Mac & Linux Standalone

Target Platform	Mac OS X
Development Build	☐
Autoconnect Profiler	☐
Script Debugging	☐
Scripts Only Build	☐
Compression Method	Default


Color Space*	Gamma
Auto Graphics API for Windo	✓
Auto Graphics API for Mac	✓
Auto Graphics API for Linux	✓
Color Gamut For Mac*	sRGB
Metal Editor Support* (Expe	☐
Metal Restricted Backbuffer	☐
Static Batching	✓
Dynamic Batching	✓
GPU Skinning*	☐
Graphics Jobs (Experimenta	☐
Lightmap Encoding	Normal Quality

Virtual Reality moved to XR Settings

Mac App Store Options

Bundle Identifier	unity.Agora Lab, Inc.Hello Gaming Vid
Version*	2.9.1.42
Build	0
Category	public.app-category.games
Mac App Store Validation	☐

Configuration

Scripting Runtime Version*	Stable (.NET 3.5 Equivalent)
Scripting Backend	Mono
Api Compatibility Level*	.NET 2.0 Subset
Disable HW Statistics*	☐
Scripting Define Symbols*	
Active Input Handling*	Input Manager

MacOS Build Settings

Codesigning for Notarization

For Notarization validation, Agora library frameworks need to be code-signed with the main build executable together. The SDK provided helper scripts to get your signing step more convenient. See instructions below.

Codesign instruction after mac build

use scripts provided in Unity SDK project folder: **Assets/AgoraEngine/Scripts/AgoraTools**

Assume the project folder is ${project_path}, your build is output to ${build_folder}, your app is TestMAC.app

Steps:

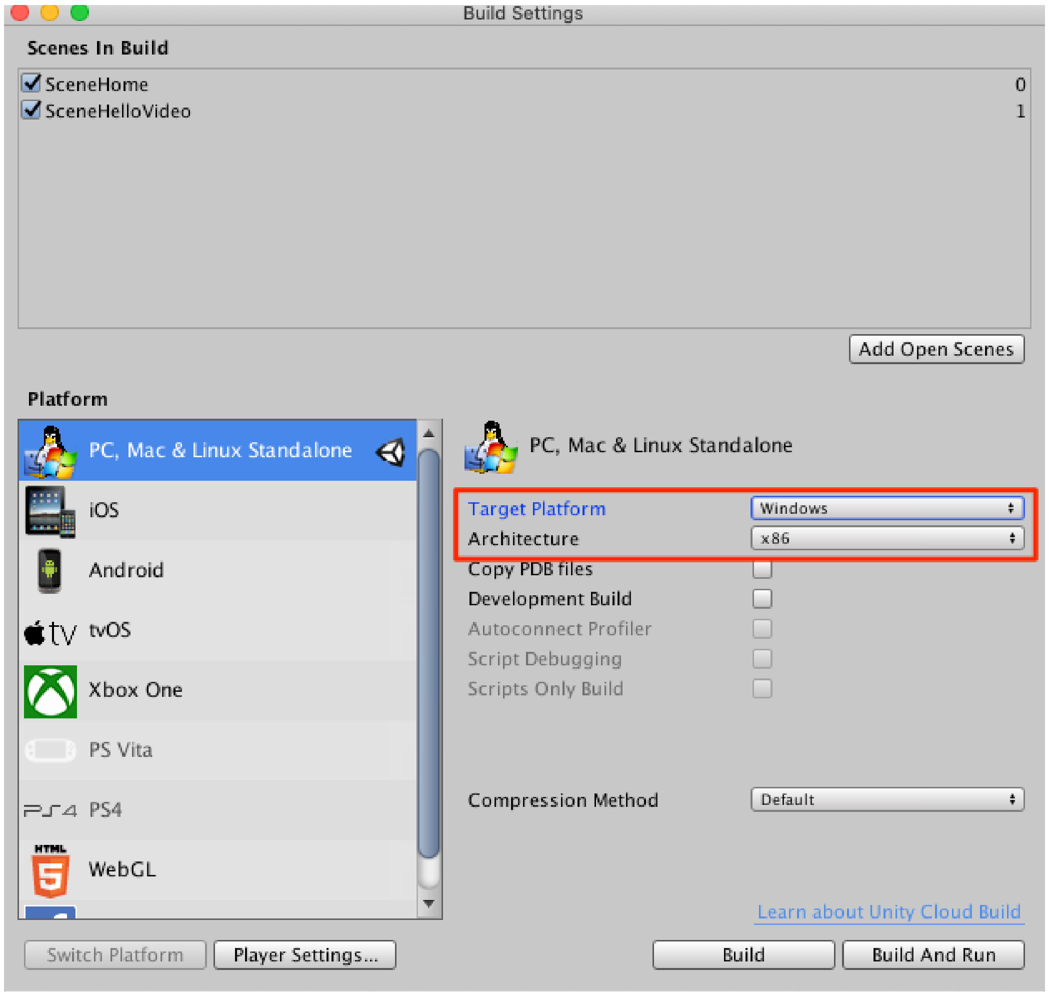
1. cd ${build_folder}
2. ${project_path}/Assets/AgoraEngine/Scripts/AgoraTools/prep_codesign.sh TestMAC.app
3. find the signature on the Mac with this command /usr/bin/security find-identity -v -p codesigning （hint from the script too）
4. (export SIGNATURE="<your signature>"; ${project_path}/Assets/AgoraEngine/Scripts/AgoraTools/signcode.sh TestMAC.app)
5. Verify that the result is

TestMac.app: valid on disk

TestMac.app: satisfies its Designated Requirement

Windows Build

Select **"PC, Mac & Linux Standalone"** from the platform list and click Switch Platform. Then Select "**Windows**" for the Target Platform. Also choose x86 for 32 bit or x86_64 for 64 bit architecture according to your Application target.



Scenes In Build	
✓ SceneHome	0
✓ SceneHelloVideo	1

Platform list: PC, Mac & Linux Standalone; iOS; Android; tvOS; Xbox One; PS Vita; PS4; WebGL

PC, Mac & Linux Standalone

Target Platform	Windows
Architecture	x86
Copy PDB files	☐
Development Build	☐
Autoconnect Profiler	☐
Script Debugging	☐
Scripts Only Build	☐
Compression Method	Default

Windows Build

Remember to set the appropriate Plugin library identities as described in the earlier section of this README file.

Conclusion

After the configuration has been set up according to the above rules, you should be able to build to the target platform. You can then talk to the other device using the Agora Voice SDK.

Other Resources

- GitHub demos: https://github.com/AgoraIO/Agora-Unity-Quickstart
- The complete API documentation is available in the Document Center.
- For technical support, submit a ticket using the Agora Dashboard or
- Join our slack community: https://bit.ly/39j4I5h
- Help each other at https://agoraiodev.slack.com/messages/unity-help-me
- Developer relations team: devrel@agora.io
- Release note:
https://docs.agora.io/en/Interactive%20Broadcast/release_unity_video?platform=Unity