

游戏多媒体引擎 基础功能开发指南 产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

基础功能开发指南

鉴权密钥

实时语音角色设置

音质选择

基础功能开发指南

鉴权密钥

最近更新时间：2024-01-18 11:56:47

为方便开发者调试和接入腾讯云游戏多媒体引擎产品，这里向您介绍适用于所有平台的鉴权密钥相关技术文档。

语音密钥后台部署

腾讯云游戏多媒体引擎提供鉴权密钥，用于实时语音及离线语音的鉴权，此文档为后台部署方案。

鉴权所用到的签名生产过程涉及到**明文**、**密钥**和**算法**。

明文

明文为以下字段的网络序拼接：

字段描述	类型/长度	值定义/备注
cVer	unsigned char(1)	版本号，填写数值：1
wOpenIDLen	unsigned short(2)	用户的帐号长度
strOpenID	string	用户的帐号字符
dwSdkAppid	unsigned short(4)	开发者的 SDKappid
dwReserved1	unsigned int(4)	填写数值：0
dwExpTime	unsigned int(4)	过期时刻（当前时间+有效期[单位：秒，建议300秒]）
dwReserved2	unsigned int(4)	填写数值：-1或者 0xFFFFFFFF
dwReserved3	unsigned int(4)	填写数值：0
wRoomIDLen	unsigned short(2)	用户要进入的房间号码长度，如果是离线语音服务，请填写0
strRoomID	string	用户要进入的房间号码字符

密钥

腾讯云 GME 控制台获取相关权限密钥。

算法

TEA 对称加密算法。

总体建议，建议接入初期使用客户端部署方案，后面可优化为部署在游戏 App 后台。

方案	优点	缺点
后台部署	安全性高	需要后台开发联调
客户端部署	接入快速	安全性低

后台部署使用方法

后台生成加密串后，下发给客户端，客户端适用于如下场景，调用 API:EnterRoom 进行进房操作时，将加密串传给进房参数的 authBuffer 字段。

算法加密详情

密钥：APPID 对应的鉴权密钥

加密算法：TEA 加密

注意：

控制台修改密钥后，15分钟 - 1小时内生效，不建议频繁更换。

加密方法

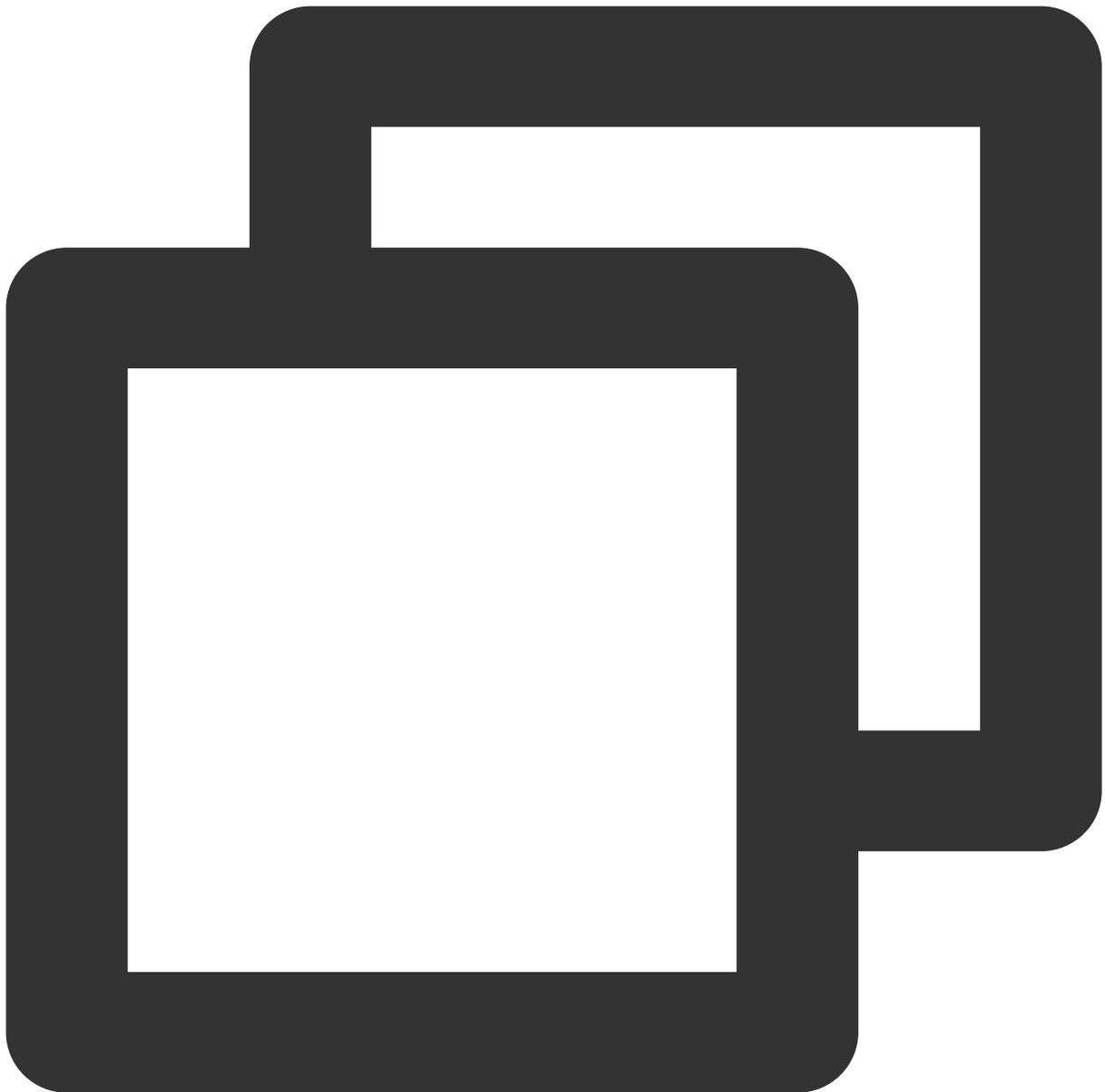
1. 把明文中的数字转换成网络字节序。
2. 把明文按顺序（明文字段声明的顺序）拼接成一个字符串。
3. 用 tea 加密拼接后的字符串，symmetry_encrypt 函数输出的字符串即为权限加密串。

注意：

不用要把二进制串转成16进制。

示例代码

以 C++ 语言为例，鉴权密钥示例代码如下：



```
unsigned char pInBuf[512]={0};
xel::byte_writer bw(pInBuf, sizeof(pInBuf));

char cVer = 1;
unsigned short wOpenIDLen = (unsigned short)strlen((const char *)strOpenID);
    if (wOpenIDLen > 127) wOpenIDLen = 127;
unsigned short wRoomIDLen = (unsigned short)strlen((const char *)strRoomID);
    if (wRoomIDLen > 127) wRoomIDLen = 127;

bw.write_byte(cVer);
bw.write_int16(wOpenIDLen);
```

```
bw.write_bytes(strOpenID, wOpenIDLen);
bw.write_int32(dwSdkAppId);
bw.write_int32(0 /*dwRoomID*/);
bw.write_int32(expTime);
bw.write_int32(nAuthBits);
bw.write_int32(0 /*dwAccountType*/);
bw.write_int16(wRoomIDLen);
bw.write_bytes(strRoomID, wRoomIDLen);

int pInLen = bw.bytes_write();

    unsigned char pEncryptOutBuf[512] = { 0 };
int iEncriptyLen = 0;

    symmetry_encrypt((const unsigned char*)pInBuf, pInLen, (const unsigned char
```

我们也提供了 Java 和 Go 语言的示例代码, [点击下载 >>](#)

实时语音角色设置

最近更新时间：2023-05-19 15:11:55

注意：GME 3.x 版本暂不支持实时语音角色设置功能。

为方便开发者调试和接入腾讯云游戏多媒体引擎产品 API，本文向您介绍游戏多媒体引擎国战语音模式的接入流程。

使用场景

在国战游戏场景中，GME 提供主播以及听众两种角色，如果在进房前设置为主播，则进房之后可以打开麦克风发言以及打开扬声器听房间内的交流声音；如果使用听众角色进入房间，进房之后即使打开了麦克风也无法在房间内发言。

前提条件

已完成 GME 应用创建，并获取 SDK AppID 和 Key。请参考 [服务开通指引](#)。

已开通 **GME 实时语音服务**。请参考 [服务开通指引](#)。

已完成 **GME SDK 接入**。请参考 [SDK 快速接入](#)。

接入步骤

接入国战语音模式流程如下：

1. [使用 GME 服务](#)
2. [设置角色](#)
3. [使用实时语音服务](#)
4. [打开麦克风](#)
5. [改变角色](#)
6. [退出房间](#)

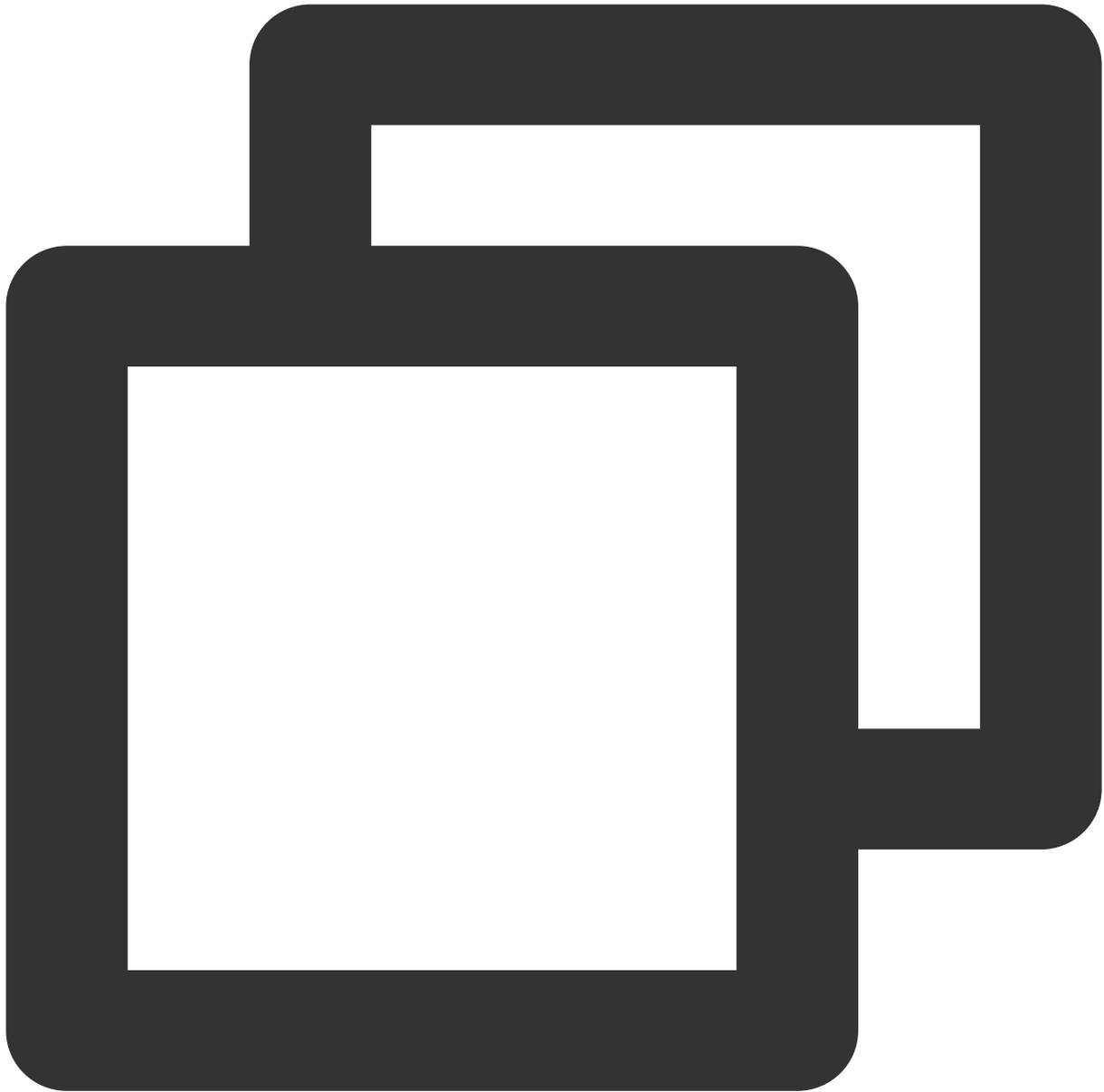
步骤1：使用 GME 服务

GME SDK 的调用和接入请参见 [Native SDK 快速接入](#)、[Unity SDK 快速接入](#)、[Unreal SDK 快速接入](#)。

步骤2：设置角色

在进房 EnterRoom 接口调用之前，需要调用设置角色接口，设置本端在实时语音中的角色。

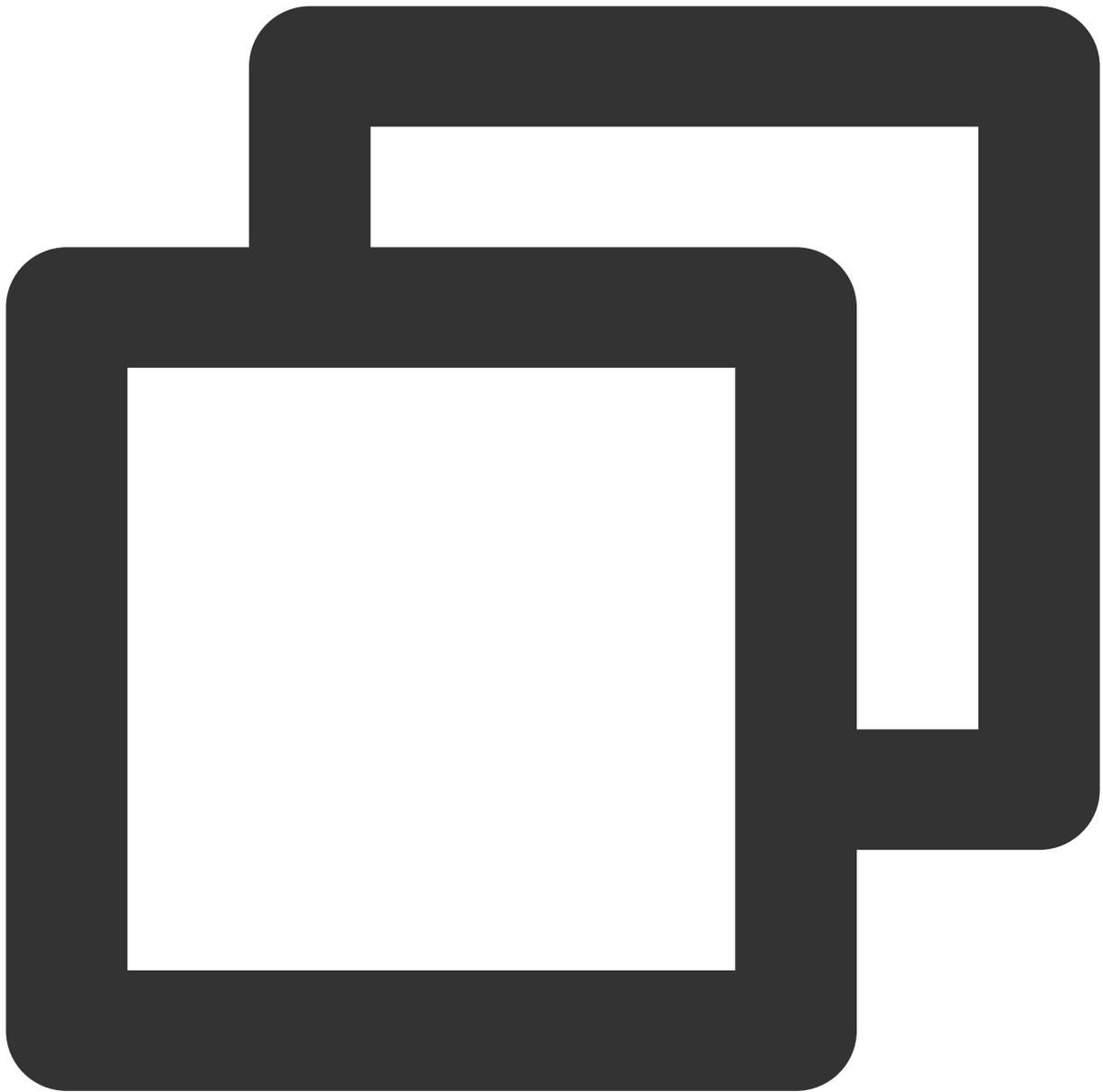
函数原型



```
public abstract int SetAudioRole(ITMG_AUDIO_MEMBER_ROLE role);
```

参数	类型	含义
role	ITMG_AUDIO_MEMBER_ROLE	ITMG_AUDIO_MEMBER_ROLE_ANCHOR 代表主播，在房间内可以打开麦克风及扬声器 ITMG_AUDIO_MEMBER_ROLE_AUDIENCE 代表听众，在房间内只能打开扬声器收听

示例代码



```
ITMGContext.GetInstance().SetAudioRole(ITMG_AUDIO_MEMBER_ROLE.ITMG_AUDIO_MEMBER_ROL
```

步骤3：使用实时语音服务

调用 `EnterRoom` 接口进入实时语音房间。

步骤4：打开麦克风

如果角色是主播，可以正常调用 `EnableMic` 接口以及 `EnableSpeaker` 接口打开麦克风以及扬声器。

如果角色是听众，可以正常使用 `EnableSpeaker` 接口打开扬声器，但是调用 `EnableMic` 接口时会返回 `AV_ERR_INVALID_ARGUMENT`（1004）错误码，提示此时是听众模式，打开麦克风无效。

步骤5：改变角色

在房间中可以调用 `SetAudioRole` 改变角色，如果此时：

未设置角色，则会转变成新设置的角色。

已经设置了角色，则会转变为新设置的角色。

如果此时未设置角色或者是主播角色，已经开启了麦克风正常通话，改变为听众角色后，麦克风依然是打开状态，此时建议业务层调用 `EnableMic` 接口修改麦克风状态，同时修改麦克风 UI 状态。

步骤6：退出房间

调用 `ExitRoom` 接口退出实时语音房间后，角色状态失效，需要重新设置角色。

音质选择

最近更新时间：2024-01-18 11:59:00

为方便开发者接入腾讯云游戏多媒体引擎产品，本文主要为您介绍适用于游戏多媒体引擎 SDK 的音质选择文档。

音质类型介绍

音质类型	含义	参数	码率	采样率	音量类型	适用场景
ITMG_ROOM_TYPE_FLUENCY	流畅音质	1	30kbps	16k	扬声器：通话音量 耳机：媒体音量 蓝牙耳机：HFP 协议，耳机采集	流畅优先、超低延迟实时语音，应用在游戏内开黑场景，适用于 FPS、MOBA 等类型的游戏
ITMG_ROOM_TYPE_STANDARD	标准音质	2	64kbps	48k	扬声器：通话音量 耳机：媒体音量 蓝牙耳机：HFP 协议，耳机采集	音质较好，延时适中，适用于狼人杀、棋牌等休闲游戏的实时通话场景
ITMG_ROOM_TYPE_HIGHQUALITY	高清音质	3	64kbps	48k	扬声器：媒体音量 耳机：媒体音量 蓝牙耳机：a2dp 协议，手机采集	超高音质，延时相对大一些，适用于音乐舞蹈类游戏以及语音互动类 App；适用于播放音乐、线上 K 歌等有高音质要求的场景

注意：

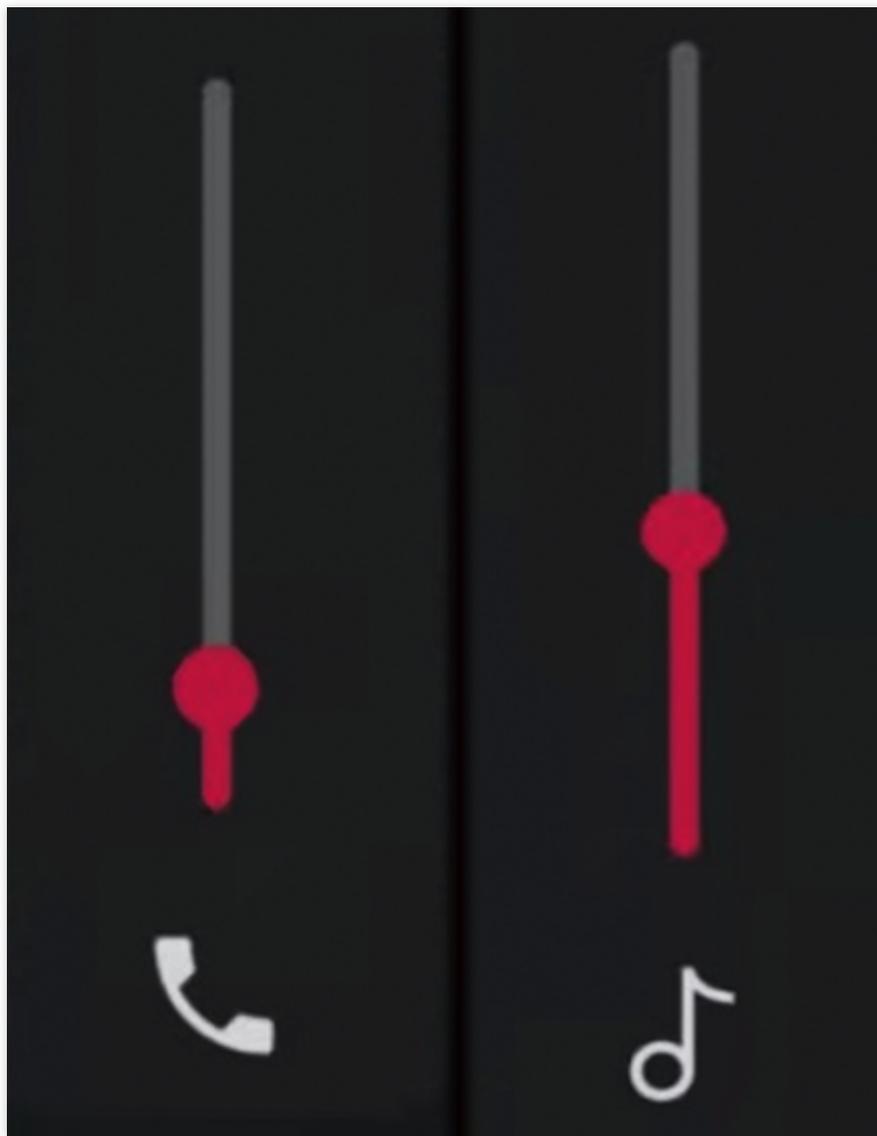
2.9.0及以上版本，Unity、UnrealEngine、Cocos SDK 如需使用标准音质以及高清音质，请参见 [升级指引](#) 进行接入。

相关概念介绍

媒体音量与通话音量

手机音量有两种模式，一种是媒体音量，一种是通话音量。媒体音量一般用于播放媒体文件，通话音量一般用于打电话、通信等场景。

以 Android 手机为例，按下手机物理音量键，会显示此时的音量类型。下图中，左侧为通话音量类型，右侧为媒体音量类型。



媒体音量与通话音量相关问题

[进房后，手机音量变得很小，开启麦克风后音量又变得很大，怎么解决？](#)

蓝牙耳机协议

蓝牙耳机协议分为以下两种，不同的蓝牙耳机协议对应的音频表现不同：

协议名称	播放表现	采集表现
HFP 协议	耳机声音只有单声道	耳机可以采集到声音
a2dp 协议	耳机声音具有双声道，音色更好，音质更高	耳机无法使用此通道进行采集，需要使用手机或者电脑自带麦克风进行采集

流量消耗

流量和码率有关，也和房间内通话人数有关。具体计算公式为： $\text{码率} \times \text{人数} / 8 = \text{字节}$ 。

音频处理效果

音频信号被其移动终端采集模块采集，经过混音消除、降噪、自动增益控制等音频的前处理过程，由音频编码器进行编码。其中前处理过程，Acoustic Echo Cancelling (AEC)、Automatic Gain Control (AGC)、Automatic Noise Suppression (ANS，也被称为 noise cancellation、noise suppression) 的使用方式，俗称 3A。

流畅音质和其他两个音质的不同在于 ANS 降噪，流畅音质下会打开 ANS，标准音质及高清音质下会关闭 ANS。