

Game Multimedia Engine

基本機能開発ガイド

製品ドキュメント



Tencent Cloud

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

カタログ：

基本機能開発ガイド

Authentication Key

リアルタイム音声ロール設定

音質設定

基本機能開発ガイド

Authentication Key

最終更新日：：2024-01-18 11:56:22

This document describes the authentication key for all platforms to help you integrate and debug GME.

Backend Deployment of Voice Key

GME provides authentication keys for voice chat and offline voice. This document describes the backend deployment scheme.

The generation process of the signature used for authentication involves **plaintext**, **secret key** and **algorithm**.

Plaintext

The plaintext is the concatenation of the following fields in endian byte order:

Field	Type/Length	Description
cVer	unsigned char(1)	Version number. Enter <code>1</code> .
wOpenIDLen	unsigned short(2)	User account length
strOpenID	string	User account's characters
dwSdkAppid	unsigned short(4)	<code>SDKappid</code> of the developer
dwReserved1	unsigned int(4)	Enter <code>0</code> .
dwExpTime	unsigned int(4)	Expiration time (current time + validity period) in seconds. <code>300</code> is recommended.
dwReserved2	unsigned int(4)	Enter <code>-1</code> or <code>0xFFFFFFFF</code> .
dwReserved3	unsigned int(4)	Enter <code>0</code> .
wRoomIDLen	unsigned short(2)	Length of the ID of the room to enter. Enter <code>0</code> for the voice messaging service.
strRoomID	string	Characters of the ID of the room to enter

Key

Get the relevant permission key in the GME console.

Algorithm

The Tiny Encryption Algorithm (TEA) symmetric algorithm is used.

Generally, we recommend that you use the client deployment scheme in the initial stage, which later can be optimized for deployment on the game application's backend.

Scheme	Pros	Cons
Backend deployment	High security	Backend development and joint testing required
Client deployment	Quick integration	Low security

Backend Deployment

The encrypted string generated on the backend is sent to the client and used for the following scenario: When the EnterRoom API is called for entering a room, the encrypted string will be transferred to the `authBuffer` field in the parameters for room entering.

Algorithm Encryption Details

Key: Authentication key of `APPID`.

Encryption algorithm: TEA.

Note:

Change of the key in the console takes effect within 15 minutes to 1 hour. We recommend that you not change it frequently.

Encryption method

1. Reorganize the numbers in the plaintext in endian order.
2. Concatenate the plaintext fields into a string in the sequence how they are declared.
3. Encrypt the concatenated string with TEA. The string output by the `symmetry_encrypt` function is the permission encryption string.

Note:

Do not convert a binary string into a hexadecimal one.

Sample code

Taking C++ as an example, below is the sample code of the authentication key:



```
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 iEncryptLen = 0;

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

You can also download the sample code in Java and Go [here](#).

リアルタイム音声ロール設定

最終更新日：2024-01-18 15:43:38

ご注意：

GME 3.xバージョンは現在、リアルタイム音声ロール設定機能をサポートしていません。

開発者がTencent Cloud GME製品のAPIを容易にデバッグして導入するために、このドキュメントではGMEのコマンドモードの導入手順を紹介します。

シナリオ

ウォーゲームのシーンで、GMEはキャスターとオーディエンスの2種類のロールを提供します。ルームに参加する前にキャスターを設定した場合、ルームに参加した後マイクをオンにして発言したり、スピーカーをオンにしてルーム内のコミュニケーション音声を聞くことができます。オーディエンスとしてルームに参加した場合、ルームに参加した後マイクをオンにしてもルーム内で発言することができません。

前提条件

GMEアプリケーションの作成が完了し、SDK AppIDとKeyを取得しました。[サービス有効化ガイド](#)をご参照ください。

リアルタイム音声サービスを有効にしました。[サービス有効化ガイド](#)をご参照ください。

GME SDKの導入が完了しました。[SDKの快速導入](#)をご参照ください。

統合の手順

ウォーゲーム音声モードを導入する手順は次の通りです：

- [GMEサービスの使用](#)
- [ロールの設定](#)
- [リアルタイム音声サービスの使用](#)
- [マイクオン](#)
- [ロールの変更](#)
- [ルーム退出](#)

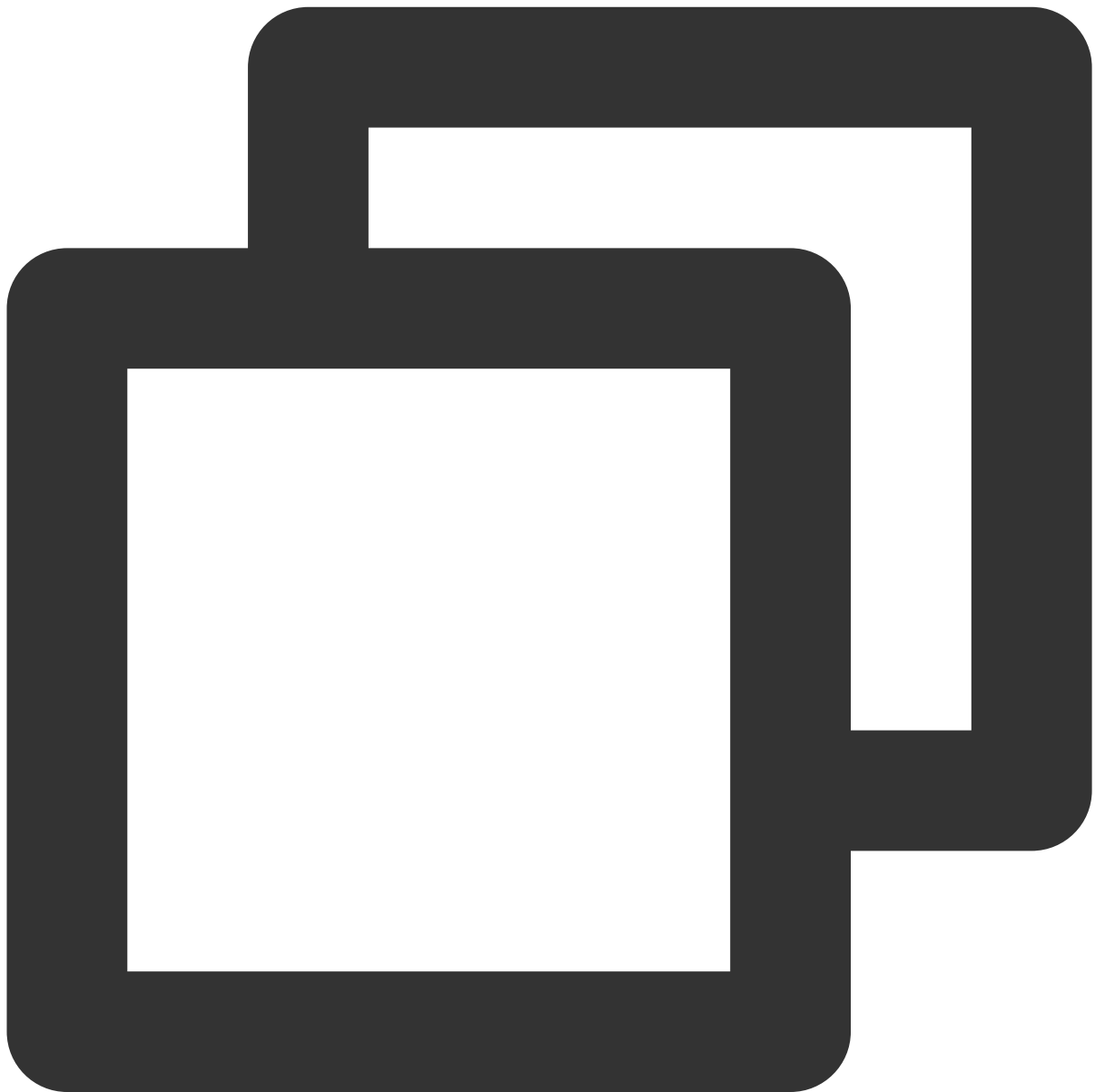
ステップ1：GMEサービスの利用

GME SDKの呼び出しと導入については、[Native SDKクイックスタート](#)、[Unity SDKクイックスタート](#)、[Unreal SDKクイックスタート](#)をご参照ください。

ステップ2：ロールの設定

ルーム参加EnterRoomインターフェースを呼び出す前に、ロール設定インターフェースを呼び出して、リアルタイム音声でローカル側のロールを設定する必要があります。

関数のプロトタイプ

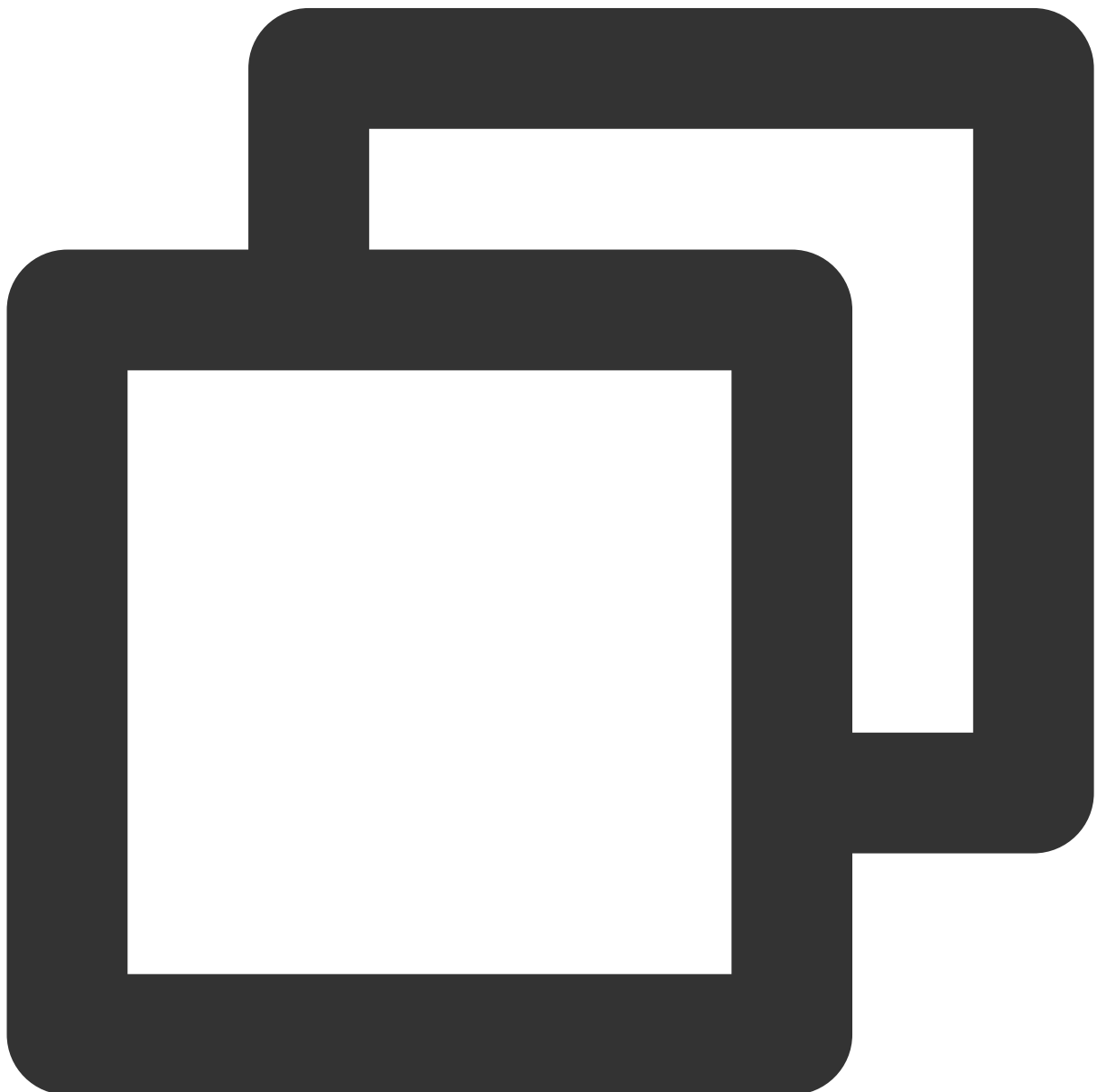


```
public abstract int SetAudioRole(ITMG_AUDIO_MEMBER_ROLE role);
```

パラメータ	タイプ	意味

タ		
role	ITMG_AUDIO_MEMBER_ROLE	ITMG_AUDIO_MEMBER_ROLE_ANCHORはキャスターを表し、ルーム内でマイクとスピーカーをオンにすることができます ITMG_AUDIO_MEMBER_ROLE_AUDIENCEはオーディエンスを表し、ルーム内でスピーカーのみをオンにして聞くことができます

サンプルコード



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

ステップ3：リアルタイム音声サービスの利用

EnterRoomインターフェースを呼び出して、リアルタイム音声ルームに入ります。

ステップ4：マイクをオンにする

ロールがキャスターの場合は、EnableMicインターフェースとEnableSpeakerインターフェースを正常に呼び出してマイクとスピーカーをオンにすることができます。

ロールがオーディエンスの場合、EnableSpeakerインターフェースを正常に使用してスピーカーをオンにすることができますが、EnableMicインターフェースを呼び出すとAV_ERR_INVALID_ARGUMENT（1004）エラーコードが返され、この時点ではオーディエンスモードであり、マイクのオンは無効であることが表示されます。

ステップ5：ロールの切り替え

ルーム内でSetAudioRoleを呼び出してロールを変更することができます。

ロールが設定されていない場合、新しく設定されたロールに切り替えます。

ロールが設定された場合は、新しく設定されたロールに切り替えます。

この時点でロールが設定されていない場合、またはキャスターである場合、通常の通話が行われるようマイクがオンの状態で、オーディエンスのロールに切り替えると、マイクがオンのままです。この場合、業務レベルではEnableMicインターフェースを呼び出してマイクの状態とマイクUIの状態を変更することをお勧めします。

手順6：ルーム退出

ExitRoomインターフェースを呼び出してリアルタイム音声ルームを終了すると、ロール状態が無効になり、ロールを再設定してください。

音質設定

最終更新日：2024-01-18 15:43:38

開発者がTencent Cloud Gaming Multimedia Engine製品のアクセスを手軽に実行できるように、このドキュメントで、Gaming Multimedia Engine SDKに向ける音質選択ドキュメントについて説明させていただきます。

音質タイプの説明

音質タイプ	意味	パラメータ	コードビット	サンプリング率	音量タイプ	適用シナリオ
ITMG_ROOM_TYPE_FLUENCY	滑らかな音質	1	30kbps	16Kス	ピーカ：通話ボリューム ヘッドセット：メディアボリューム Bluetooth ヘッドセット：HFPプロトコル、ヘッドセット収集	スムーズさを優先に超低遅延のリアルタイムボイスでFPS、MOBAなどのようなゲーム内のボイスシナリオに適用されます。
ITMG_ROOM_TYPE_STANDARD	標準音質	2	64kbps	48k	スピーカ：通話ボリューム ヘッドセット：メディアボリューム Bluetooth ヘッドセット：HFPプロトコル、ヘッドセット収集	音質が良く、タイムラグも適度で、人狼ゲームやボードゲームなどのカジュアルなゲームにおけるリアルタイムな通話シナリオに適します。

ITMG_ROOM_TYPE_HIGHQUALITY	高音質	3	64kbps	48k	スピーカー：メディア音量 イヤホン：メディア音量 Bluetooth ヘッドセット：a2dpプロトコル、携帯電話収集	高音質で、遅延が比較的大きく、ミュージック・ダンスゲームおよびボイスインタラクシオンなどのAPP、音楽の再生、オンラインカラオケなど、高音質が要求されるシナリオに適します。
----------------------------	-----	---	--------	-----	---	--

ご注意：

2.9.0以上の場合、Unity、UnrealEngine、Cocos SDK標準および高品位の音質を使用するには、[アップグレードガイド](#)をご参照ください。

関連概念の紹介

メディア音量と通話音量

スマホの音量には、メディアの音量と通話の音量の2パターンがあります。メディア音量はメディアファイルの再生に、通話音量は電話や通信などに使われます。

Androidスマホの場合、スマホの物理音量ボタンを押すと、そのときの音量タイプが表示されます。次の図に示すように、左側が通話音量のタイプ、右側がメディア音量のタイプです。



媒体音量与通话音量相关问题

ルームに入ると、携帯電話の音量が小さくなります。マイクをオンにすると音量が大きくなりますが、どのように対処しますか。

Bluetoothヘッドフォンプロトコル

Bluetoothヘッドセットプロトコルには次の2種類があります。オーディオパフォーマンスはBluetoothヘッドセットプロトコルによって異なります。

プロトコル名	再生パフォーマンス	収集パフォーマンス
HFPプロトコル	ヘッドフォン音声はモノラルだけ	ヘッドフォンが集音可能
a2dp	ヘッドフォンの音は2チャンネル	ヘッドフォンはこのチャンネルを使用して収集することができ

プロトコル	ネル、音色がより良く、音質がより高い	なく、携帯電話またはパソコンのマイクを使用して収集する必要があります
-------	--------------------	------------------------------------

トラフィック消費量

通信量はコードビットと関係があり、ルームの中で通話している人の数とも関係があります。具体的な計算式は次のとおりです： $\text{コードビット} \times \text{人数} / 8 = \text{バイト}$ 。

オーディオ処理効果

オーディオ信号はそのモバイル端末収集モジュールによって収集され、ミックス除去、ノイズ低減、自動利得制御などのオーディオの前処理過程を経て、オーディオエンコーダによって符号化されます。このうち前処理過程は、Acoustic Echo Cancelling (AEC)、Automatic Gain Control (AGC)、Automatic Noise Suppression (ANS、noise cancellation、noise suppressionとも呼ばれる)の使用方式は俗に3Aと呼ばれます。

スムーズな音質と他の2つの音質との違いは、ANSのノイズ低減です。スムーズな音質の場合はANSがオンになり、標準およびHDの音質の場合はANSがオフになります。