Game Multimedia Engine

클라이언트 API

제품 문서

목록:

클라이언트 API
Unity SDK
SDK 통합

최종 업데이트 날짜: : 2024-01-18 16:05:38

본문은 Unity 개발자를 위해 Tencent Cloud Game Multimedia Engine(GME) API용 Unity 프로젝트를 구성하는 방법을 안내합니다.

SDK 다운로드

1. 해당 Demo 및 SDK를 다운로드하십시오. 자세한 내용은 SDK 다운로드 가이드를 참고하십시오.
2. 페이지에서 Unity용 SDK 리소스를 찾습니다.
3. **다운로드**를 클릭합니다. 압축 해제 후 다운로드한 SDK 리소스에는 다음 파일이 포함됩니다.

파일 이름	설명	용도
Plugins	SDK 라이브러리 파일	플랫폼별 라이브러리 파일 저장
GMESDK	SDK 코드 파일	API 제공

4. HD 음질을 사용하려면 Using HD Sound Quality를 참고하십시오.

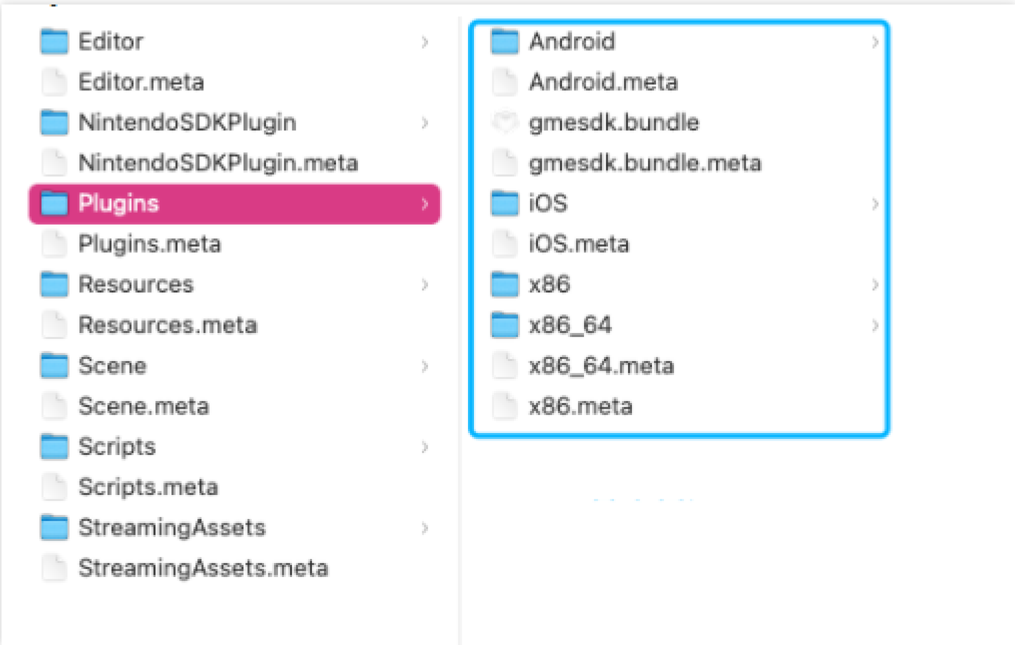
지원되는 플랫폼

Unity용 SDK는 Windows, Mac, Android, iOS, PlayStation, Xbox, Switch 및 WebGL 플랫폼 아키텍처를 동시에 통합했습니다.

프로젝트 구성

1단계: Plugins 파일 가져오기

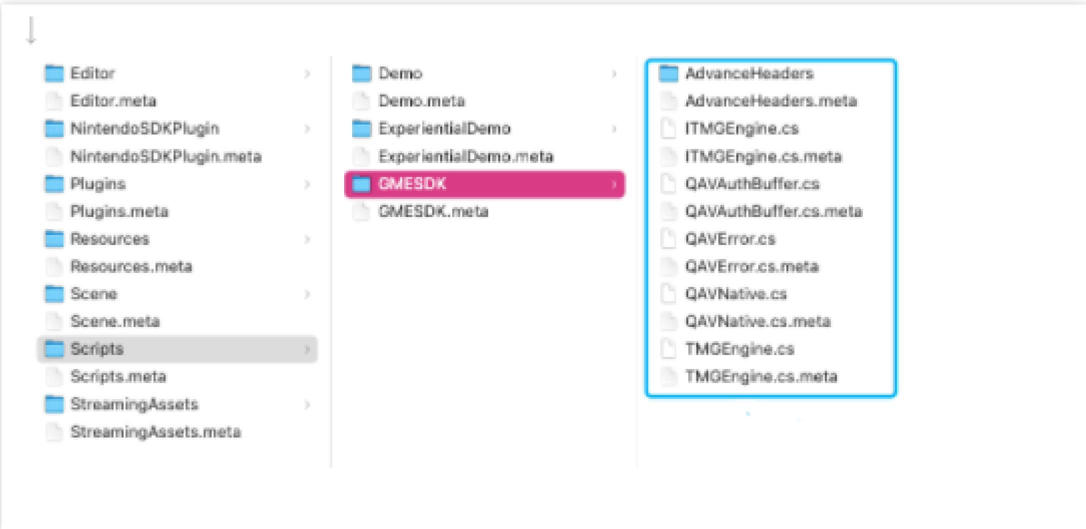
아래와 같이 SDK의 Plugins 폴더에서 **Unity 프로젝트**>**Assets**>**Plugins** 아래의 폴더로 파일을 복사합니다.



설명 :
win32 아키텍처에서 실행 파일을 내보낼 필요가 없으면 Plugins 폴더 아래의 x86 폴더를 삭제하십시오.

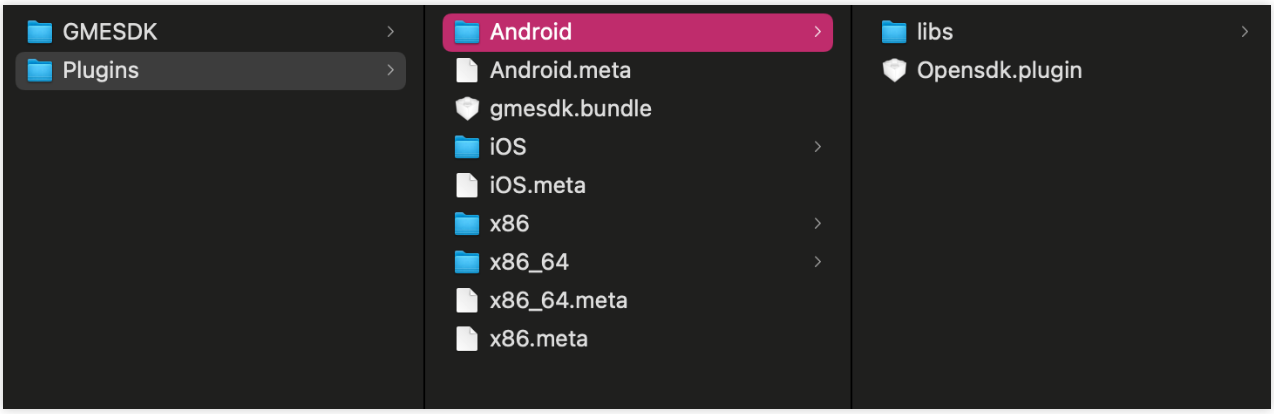
2단계: 코드 파일 가져오기

아래와 같이 SDK의 Scripts 폴더에 있는 파일을 Unity 프로젝트의 코드를 저장하는 데 사용되는 폴더에 복사합니다.



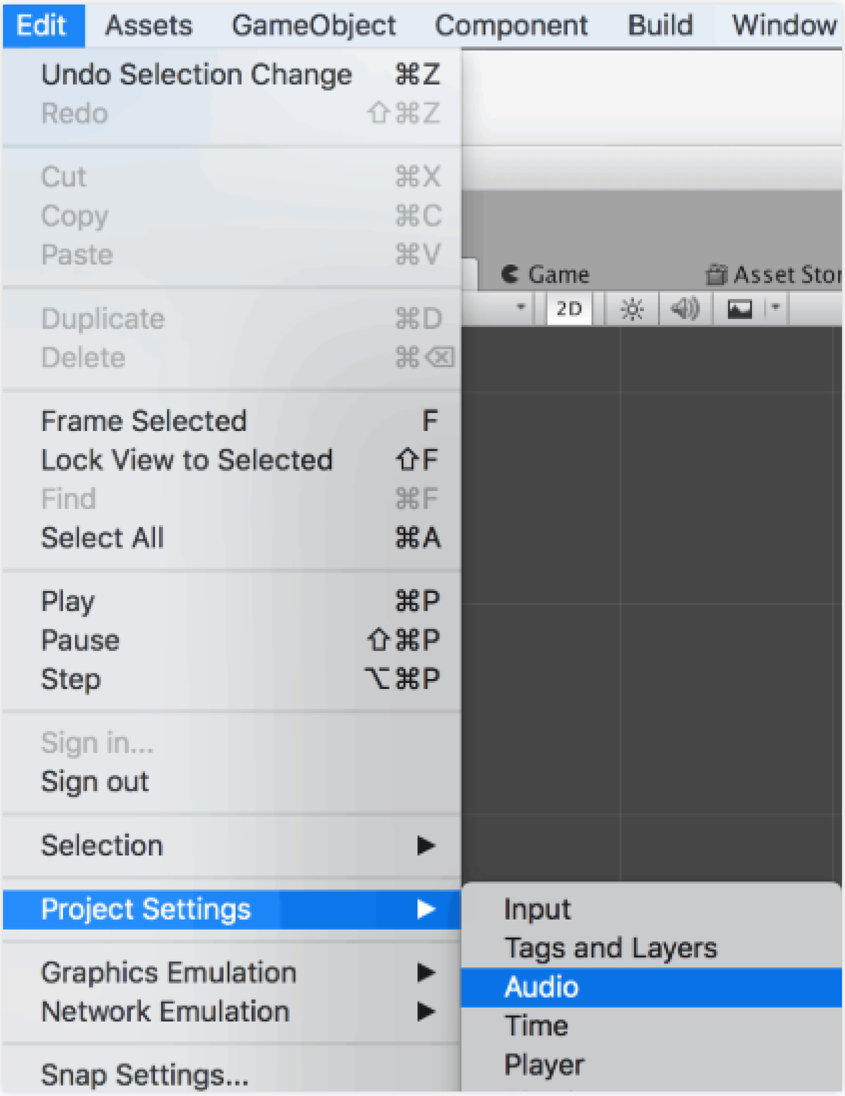
Unity 2021 구성

Unity Editor 2021 이상을 사용하는 경우 Plugins > Android > Opensdk.plugin 아래의 lib 폴더를 잘라내기 하고 Opensdk.plugin과 동일한 수준의 프로젝트의 Plugins 파일의 Android 디렉터리에 붙여넣어야 합니다.



오디오 설정

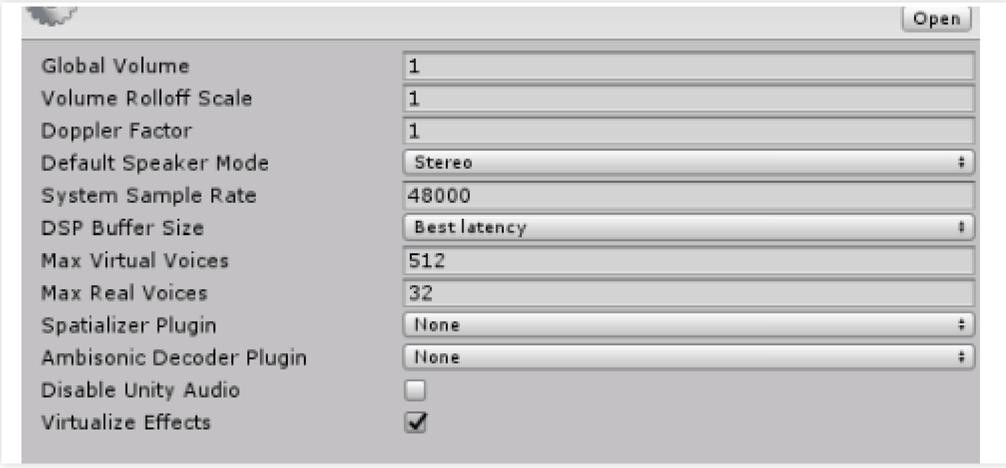
Unity 편집기에서 **Edit**>**Project Setting**>**Audio**로 이동하고 기본 시스템 설정을 사용합니다. 설정을 변경하면 아래와 같이 iOS 장치에 설정된 하드웨어 버퍼로 인해 Unity 재생 사운드 효과가 영향을 받습니다.



Unity Audio Setting

Project Setting에서 Audio 모듈을 설정하지 마십시오.

설정이 다음과 같으면 iOS 장치에 설정된 하드웨어 버퍼로 인해 Unity 재생 사운드 효과가 중단됩니다.



MacOS에서의 작업

Unity를 사용하여 MacOS 10.15.x에서 GME SDK를 통합하는 경우 `com.apple.quarantine` 속성으로 인해 실행 중에 파일이 손상되었다는 오류가 표시됩니다.

가장 직접적인 해결책은 아래와 같이 `com.apple.quarantine` 속성을 삭제하는 것입니다.

1. 터미널에서 cd 명령을 실행하여 프로젝트의 `Unity_OpenSDK_Audio/Assets/Plugins/` 폴더로 이동합니다.
2. 다음 명령을 실행합니다.

```
$ xattr -d com.apple.quarantine gmesdk.bundle
```

설명 :

이 작업은 리스크가 있습니다. 액세스하려면 이전 버전의 MacOS를 사용하는 것이 좋습니다.

실시간 음성 채팅

최종 업데이트 날짜: : 2024-01-18 16:05:38

본문은 Unity 음성 채팅 기능을 위해 Tencent Cloud Game Multimedia Engine(GME) 클라이언트 API에 액세스하고 디버깅하는 방법을 설명합니다.

GME 사용을 위한 주요 고려 사항

GME는 Init 및 Poll과 같은 핵심 API에 의존하는 실시간 음성, 음성 메시지 및 음성 텍스트 변환 서비스를 제공합니다.

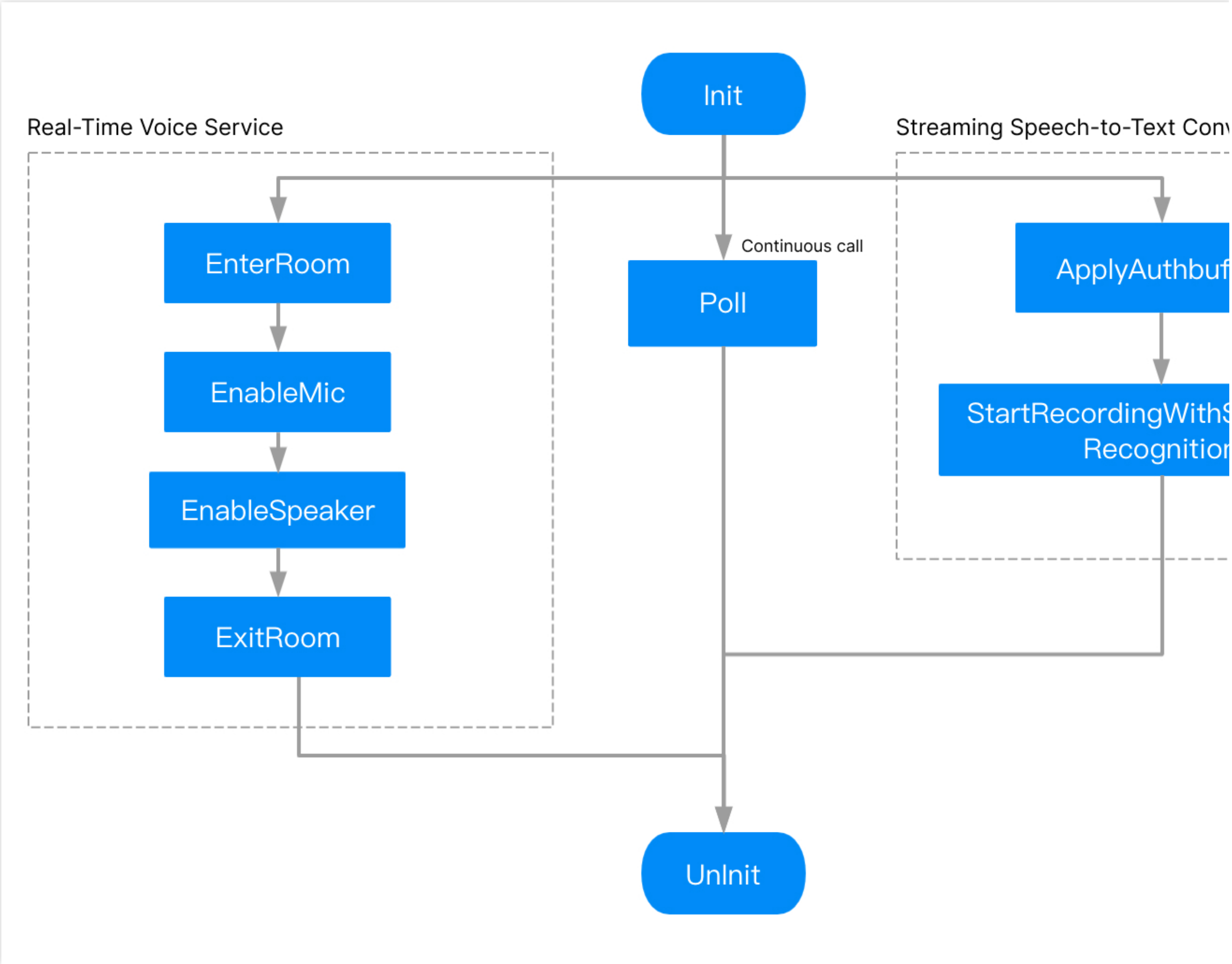
주요 사항

서비스 활성화에 설명된 대로 GME 애플리케이션을 생성하고 SDK의 AppID 및 Key를 가져옵니다.
GME 실시간 음성 채팅 서비스, 음성 메시지 서비스 및 텍스트 변환 서비스를 활성화합니다. 자세한 내용은 서비스 활성화 가이드를 참고하십시오.
GME를 사용하기 전에 프로젝트를 구성하십시오. 그렇지 않으면 SDK가 적용되지 않습니다.
GME API가 성공적으로 호출되면 QAVError.OK가 0 값으로 반환됩니다.
GME API는 동일한 스레드에서 호출되어야 합니다.
GME가 이벤트 콜백을 트리거하려면 Poll API를 주기적으로 호출해야 합니다.
에러 코드 세부 정보는 에러 코드를 참고하십시오.
GME는 Unity-WebGL 플랫폼에서 간단한 음성 채팅 기능만 지원합니다. 자세한 내용은 H5 Project Configuration을 참고하십시오.

SDK에 연결

중요 단계

SDK 연결 관련 주요 프로세스는 다음과 같습니다.



1. GME 초기화
2. 주기적으로 Poll을 호출하여 콜백 트리거
3. 음성 채팅방 입장
4. 마이크 켜기
5. 스피커 켜기
6. 음성 채팅방 나가기
7. GME 초기화 취소

C# 클래스

클래스	설명
ITMGContext	주요 API
ITMGRoom	방 API
ITMGRoomManager	방 관리 API

ITMGAudioCtrl	오디오 API
ITMGAudioEffectCtrl	음향 효과 및 반주 API

주요 API

API	API 설명
Init	GME 초기화
Poll	이벤트 콜백 트리거
Pause	시스템 일시 중지
Resume	시스템 복구
Uninit	GME 초기화 취소

헤더 파일 가져오기

```
using GME;
```

인스턴스 가져오기

QAVContext.GetInstance() 대신 ITMGContext 메소드를 사용하여 Context 인스턴스를 가져옵니다.

SDK 초기화

실시간 음성, 음성 메시지, 음성 텍스트 변환 서비스를 사용하려면 먼저 **Init API를 통해 SDK를 초기화해야 합니다**. Init API는 다른 API와 동일한 스레드에서 호출해야 합니다. 기본 스레드에서 모든 API를 호출하는 것이 좋습니다.

API 프로토타입

```
//class ITMGContext
public abstract int Init(string sdkAppID, string openID);
```

매개변수	유형	설명
sdkAppId	string	GME 콘솔에서 제공되는 AppID로, 서비스 활성화의 안내에 따라 얻을 수 있습니다.
openID	string	openID는 Int64 유형만 가능하며 string으로 변환되어 전달됩니다. 해당 규칙을 사용자 정의할 수 있으며 App에서 고유해야 합니다. Openid를 문자열로 전달하려면 Submit Ticket하여 신청하십시오.

반환된 값

반환 값	처리
QAVError.OK= 0	SDK 초기화 성공
AV_ERR_SDK_NOT_FULL_UPDATE=7015	SDK 파일이 완전한지 확인합니다. 삭제한 후 SDK를 다시 가져오는 것이 좋습니다

7015 오류 메시지

7015 에러 코드는 md5로 판단됩니다. 통합 중에 이 오류가 보고되면 메시지에 따라 SDK 파일의 무결성과 버전을 확인하십시오.

반환 값 AV_ERR_SDK_NOT_FULL_UPDATE는 사전 **알림일 뿐**이며 초기화 실패를 일으키지는 않습니다.

타사 강화, Unity 패키징 메커니즘 및 기타 요인으로 인해 라이브러리 파일의 md5가 영향을 받아 오판이 발생할 수 있습니다. **정식 출시를 위한 로직에서는 이 오류를 무시**하고 UI에 표시하지 않도록 하십시오.

예시 코드

```
int ret = ITMGContext.GetInstance().Init(sdkAppId, openID);
// 반환된 값으로 초기화 성공 여부 판단
if (ret != QAVError.OK)
    {
        Debug.Log("SDK 초기화 실패:"+ret);
        return;
    }
```

이벤트 콜백 트리거

이벤트 콜백은 update에서 Poll API를 주기적으로 호출하여 트리거할 수 있습니다. Poll API는 GME의 메시지 펌프이며 GME가 이벤트 콜백을 트리거하도록 주기적으로 호출해야 합니다. 그렇지 않으면 전체 SDK 서비스가 비정상적으로 실행됩니다. 자세한 내용은 SDK 다운로드 가이드의 EnginePollHelper 파일을 참고하십시오.

반드시 주기적으로 Poll API 호출

비정상적인 API 콜백을 방지하기 위해 Poll API는 주기적으로 메인 스레드에서 호출되어야 합니다.

API 프로토타입

```
ITMGContext public abstract int Poll();
```

예시 코드

```
public void Update()
    {
        ITMGContext.GetInstance().Poll();
```

```
    }
```

시스템 일시 중지

시스템에서 Pause 이벤트가 발생하면 엔진에도 일시 중지를 알려야 합니다. 예를 들어 애플리케이션이 백그라운드로 전환되고(OnApplicationPause, isPause=True) 방에서 오디오를 재생하기 위해 백그라운드가 필요하지 않은 경우 Pause API를 호출하여 GME 서비스를 일시 중지하십시오.

API 프로토타입

```
ITMGContext public abstract int Pause()
```

시스템 복구

시스템에서 Resume 이벤트가 발생하면 엔진에도 Resume에 대해 알려야 합니다. Resume API는 음성 채팅 복구만 지원합니다.

API 프로토타입

```
ITMGContext  public abstract int Resume()
```

SDK 초기화 해제

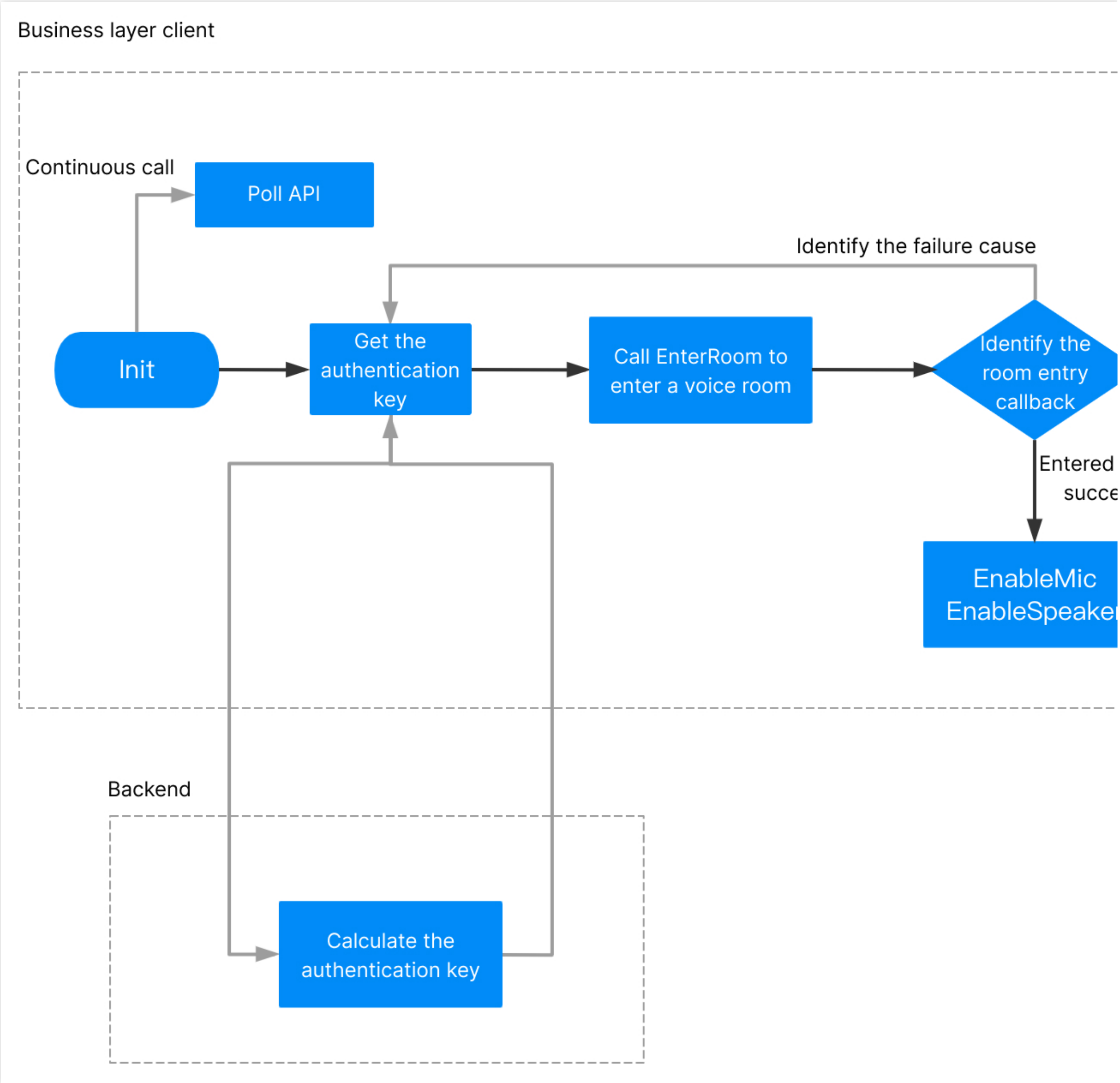
이 API는 SDK 초기화 해제를 통해 초기화를 해제하는 데 사용됩니다. **게임 비즈니스 계정이 openid에 바인딩되어 있는 경우 게임 계정을 전환하려면 GME를 초기화 해제한 다음 새 openid를 사용하여 다시 초기화해야 합니다.**

API 프로토타입

```
ITMGContext public abstract int Uninit()
```

음성 채팅방 API

음성 채팅을 시작하기 전에 초기화하고 SDK를 호출하여 방에 입장해야 합니다.
서비스 이용 중 궁금하신 사항은 실시간 음성 채팅 FAQ를 참고하십시오.



API	API 설명
GenAuthBuffer	로컬 인증 키 계산
EnterRoom	라이브 룸 입장
ExitRoom	방 퇴장
IsRoomEntered	사용자의 방 입장 여부 판단
SwitchRoom	방 전환

로컬 인증 키 계산

관련 기능의 암호화 및 인증을 위해 AuthBuffer를 생성합니다. 프로덕션 환경에서 릴리스하려면 인증 키에 설명된 대로 백엔드 배포 키를 사용하십시오.

API 프로토타입

```
QAVAuthBuffer GenAuthBuffer(int appId, string roomId, string openId, string key)
```

매개변수	유형	설명
appId	int	Tencent Cloud 콘솔의 AppId.
roomId	string	최대 127자의 방 ID.
openId	string	Init 시 openID와 동일한 사용자 ID.
key	string	Tencent Cloud 콘솔의 권한 키.

예시 코드

```
public static byte[] GetAuthBuffer(string AppID, string RoomID,string OpenId, strin
        return QAVAuthBuffer.GenAuthBuffer(int.Parse(AppID), RoomID, OpenId, AuthKe
}
```

WebGL측 적응

WebGL 플랫폼에서 로컬 인증 함수를 호출한 후 인증 값은 js 코드에 저장되며 인증된 authBuffer는 c# 레이어로 반환되지 않습니다. 사용자가 로컬 인증을 위해 GetAuthBuffer를 호출한 후, 방 입장 시 호출되는 API 매개변수에서 인증 매개변수는 공백 또는 임의의 값으로 채워집니다.
백그라운드 컴퓨팅 인증 체계를 사용하는 경우 GetAuthBuffer API를 호출할 필요가 없습니다.

방 입장

생성된 인증 정보로 방에 입장하기 위해 사용하는 API입니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되지 않습니다.

주의사항:

방 입장 콜백 result가 0이면 방 입장 성공입니다. EnterRoom API에서 0이 반환된다고 해서 반드시 방 입장이 성공한 것은 아닙니다.
방의 오디오 유형은 방에 들어오는 첫 번째 사용자에 의해 결정됩니다. 이후 방에 있는 구성원이 방 유형을 변경하면 모든 구성원에게 적용됩니다. 예를 들어, 처음 방에 입장한 사용자가 원활 음질을 사용하고 두 번째 입장한 사용자가 HD 음질을 사용했다면 두 번째 사용자의 방 오디오 유형은 원활 음질로 변경됩니다. 방의 구성원이 ChangeRoomType API를 호출한 후에만 방의 오디오 유형이 변경됩니다.

API 프로토타입

```
ITMGContext EnterRoom(string roomId, int roomType, byte[] authBuffer)
```

매개변수	유형	설명
roomId	string	최대 127자의 방 ID.
roomType	ITMGRoomType	방 유형. 게임의 경우 ITMG_ROOM_TYPE_FLUENCY를 선택하는 것이 좋습니다. 방 오디오 유형에 대한 자세한 내용은 음질 선택을 참고하십시오.
authBuffer	Byte[]	인증 코드

예시 코드

```
ITMGContext.GetInstance().EnterRoom(strRoomId, ITMGRoomType.ITMG_ROOM_TYPE_FLUENCY,
```

방 입장 콜백

사용자가 방에 들어가면 방 입장 결과가 다시 호출되어 처리를 위해 들을 수 있습니다. 성공적인 콜백은 방 입장이 성공적이며 **과금**이 시작됨을 의미합니다.

과금 참고:

구매 가이드.

과금.

음성 채팅 사용 시 클라이언트가 서버와 연결이 끊긴 경우에도 계속 과금됩니까?

API 프로토타입

```
public delegate void QAVEnterRoomComplete(int result, string error_info);
public abstract event QAVEnterRoomComplete OnEnterRoomCompleteEvent;
```

예시 코드

```
//이벤트 수신:
ITMGContext.GetInstance().OnEnterRoomCompleteEvent += new QAVEnterRoomComplete(OnEn

//수신한 이벤트 처리:
void OnEnterRoomComplete(int err, string errInfo)
    {
    if (err != 0) {
              ShowLoginPanel("에러 코드:" + err + " 에러 메시지:" + errInfo);
            return;
    }else{
        //방 입장 성공
```



```
    }
}
```

Data 상세정보

메시지	Data	예시
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

네트워크 연결이 끊어지면 연결 끊김 콜백 알림 `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` 가 발생합니다. 이때 SDK는 자동으로 다시 연결되며 콜백은 `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` 입니다. 재연결에 성공하면 `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` 콜백이 발생합니다.

에러 코드

에러 코드 값	원인 및 제안 솔루션
7006	인증 실패 원인: AppID가 존재하지 않거나 올바르지 않음 authbuff 인증 오류 인증 만료 잘못된 OpenId
7007	이미 다른 방에 있음
1001	방에 입장 중에 이 작업을 반복함. 방 입장 콜백이 반환될 때까지 방 입장 API를 호출하지 않는 것이 좋습니다.
1003	이미 방에 입장한 후에 또 다시 입장 API를 호출함
1101	SDK가 초기화되었는지, OpenId가 규칙을 준수하는지, API가 동일한 스레드에서 호출되는지, Poll API가 정상적으로 호출되는지 확인합니다

방 퇴장

이 API는 현재 방을 나가는 데 사용되며, 비동기 API입니다. 반환된 값 AV_OK는 성공적인 비동기 전달을 나타냅니다. 애플리케이션에서 방 퇴장 직후에 방 입장이 수행되는 시나리오가 있는 경우 API 호출 중에 ExitRoom API에서 RoomExitComplete 콜백 알림을 기다릴 필요가 없습니다. 대신 EnterRoom API를 직접 호출할 수 있습니다.

API 프로토타입

```
ITMGContext ExitRoom()
```

예시 코드

```
ITMGContext.GetInstance().ExitRoom();
```

방 퇴장 콜백

룸 퇴장 후 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.

API 프로토타입

```
ublic delegate void QAVExitRoomComplete();
public abstract event QAVExitRoomComplete OnExitRoomCompleteEvent;
```

예시 코드

```
이벤트 수신:
ITMGContext.GetInstance().OnExitRoomCompleteEvent += new QAVExitRoomComplete(OnExit
수신된 이벤트 처리:
void OnExitRoomComplete(){
    //방 퇴장 후 콜백 전송
}
```

사용자의 방 입장 여부 판단

이 API는 사용자가 방에 들어왔는지 여부를 확인하는 데 사용됩니다. bool 유형의 값이 반환됩니다. 방 입장 중에는 이 API를 호출하지 마십시오.

API 프로토타입

```
ITMGContext abstract bool IsRoomEntered()
```

예시 코드

```
ITMGContext.GetInstance().IsRoomEntered();
```

방 전환

사용자는 이 API를 호출하여 방 입장 후 빠르게 음성 채팅방을 전환할 수 있습니다. 방이 전환된 후 장치는 재설정되지 않습니다. 즉, 이 방에서 마이크가 이미 활성화된 경우 방이 전환된 후에도 마이크는 계속 활성화됩니다.

신속하게 방을 전환하기 위한 콜백은 ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM이며 필드는 error_info 및 result입니다.

API 프로토타입

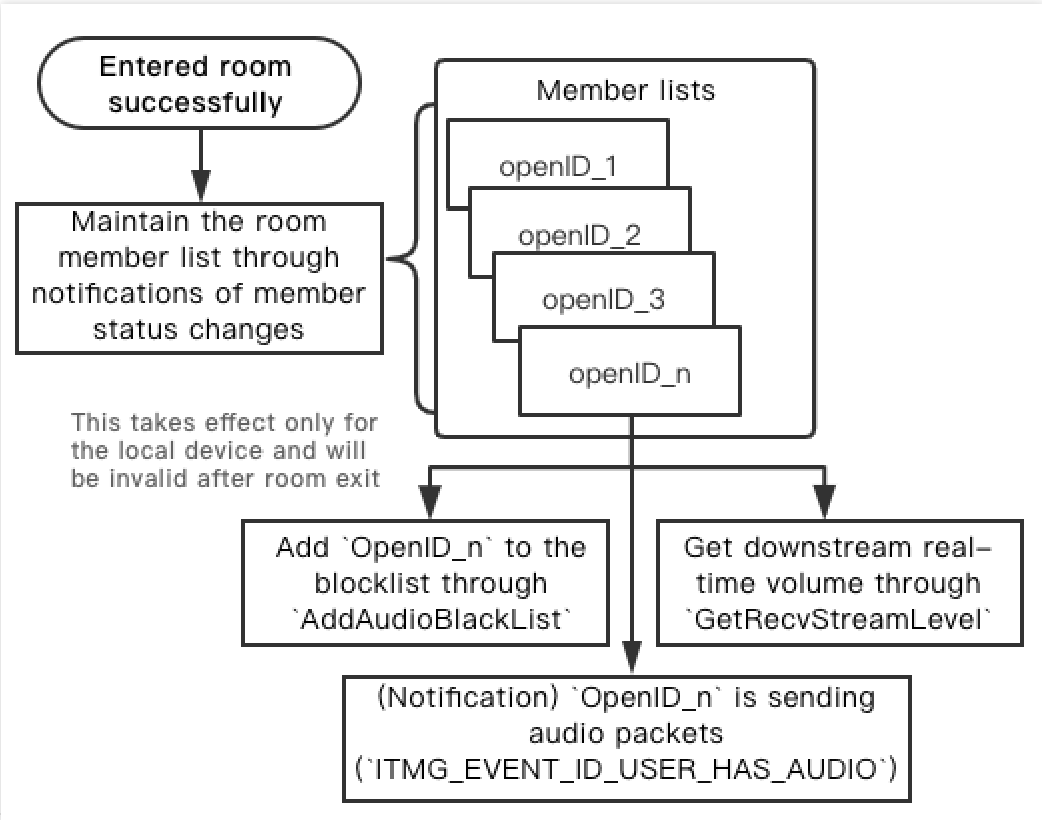
```
public abstract int SwitchRoom(string targetRoomID, byte[] authBuffer);
```

유형 설명

매개변수	유형	설명
targetRoomID	String	입장할 방 ID
authBuffer	byte[]	입장할 방의 ID로 새로운 인증 생성

방 상태 유지 보수

이 섹션의 API는 말하는 구성원과 방 입장/퇴장 구성원을 표시하고 비즈니스 레이어에서 방의 구성원을 음소거하는 데 사용됩니다.



API/알림	설명
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	구성원 상태가 변경됨

AddAudioBlackList	방에 있는 구성원 음소거
RemoveAudioBlackList	음소거 해제

구성원 방 입장 및 발언 상태 알림 이벤트

이 이벤트는 방에서 발언하는 사용자를 가져와 UI에 사용자를 표시하며, 누군가의 방 입장/퇴장 시 알림을 보내는 데 사용됩니다.

이 이벤트에 대한 알림은 상태가 변경될 때만 전송됩니다. 구성원 상태를 실시간으로 가져오려면 비즈니스 레이어에서 알림을 받을 때 캐시합니다. event_id, count 및 openIdList가 포함된 이벤트 메시지 ITMG_MAIN_EVNET_TYPE_USER_UPDATE가 반환되며 이는 OnEvent 알림에서 판단됩니다.

EVENT_ID_ENDPOINT_NO_AUDIO 오디오 이벤트 알림은 임계값을 초과한 경우에만 전송됩니다. 즉, 방에 있는 다른 구성원은 로컬 클라이언트가 2초 동안 음성을 캡처하지 않은 후에만 로컬 사용자가 발언을 중지했다는 알림을 받을 수 있습니다.

오디오 이벤트는 구성원의 발언 상태만 반환하고 특정 볼륨 수준은 반환하지 않습니다. 방에 있는 구성원의 특정 볼륨 수준이 필요한 경우 GetRecvStreamLevel API를 사용할 수 있습니다.

event_id	설명	애플리케이션측 유지 보수
EVENT_ID_ENDPOINT_ENTER	방에 입장하는 구성원의 openid 반환	구성원 목록
EVENT_ID_ENDPOINT_EXIT	방에서 퇴장하는 구성원의 openid 반환	구성원 목록
EVENT_ID_ENDPOINT_HAS_AUDIO	방에서 오디오 패킷을 보내는 구성원의 openid 반환. 이 이벤트는 사용자가 말하고 있는지 여부를 확인하고 성문 효과를 표시하는 데 사용할 수 있습니다.	채팅 구성원 목록
EVENT_ID_ENDPOINT_NO_AUDIO	방에서 오디오 패킷 전송을 중지하는 구성원의 openid 반환	채팅 구성원 목록

예시 코드

```
public delegate void QAVEndpointsUpdateInfo(int eventID, int count, [MarshalAs(Unma
public abstract event QAVEndpointsUpdateInfo OnEndpointsUpdateInfoEvent;

//이벤트 수신:
ITMGContext.GetInstance().OnEndpointsUpdateInfoEvent += new QAVEndpointsUpdateInfo(
//수신한 이벤트 처리:
void OnEndpointsUpdateInfo(int eventID, int count, string[] openIdList)
{
```

```
                //프로세스
            switch (eventID)
             {
             case EVENT_ID_ENDPOINT_ENTER:
                  //구성원 방 입장
                  break;
             case EVENT_ID_ENDPOINT_EXIT:
                  //구성원 방 퇴장
                break;
            case EVENT_ID_ENDPOINT_HAS_AUDIO:
                //구성원 오디오 패키지 발송
                break;
            case EVENT_ID_ENDPOINT_NO_AUDIO:
                //구성원 오디오 패키지 발송 중단
                break;

            default:
                break;
             }
        break;
}
```

방에 있는 구성원 음소거

이 API는 오디오 데이터 블록리스트에 ID를 추가하는 데 사용됩니다. 이 작업은 누군가의 오디오를 차단하지만 다른 장치에는 영향을 주지 않고 로컬 장치에만 적용됩니다. 반환된 값 0은 호출이 성공했음을 나타냅니다. 사용자 A, B, C가 모두 방에서 마이크를 사용하여 말하고 있다고 가정합니다.
A가 C를 차단하면 A는 B만 들을 수 있습니다.
B가 A도 C도 차단하지 않으면 B는 둘 다 들을 수 있습니다.
C가 A도 B도 차단하지 않으면 C는 둘 다 들을 수 있습니다.
이 API는 방에서 사용자가 음소거된 시나리오에 적합합니다.

API 프로토타입

```
ITMGContext ITMGAudioCtrl AddAudioBlackList(String openId)
```

매개변수	유형	설명
openId	String	블록리스트에 추가할 사용자 openid

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl ().AddAudioBlackList (openId);
```

음소거 해제

이 API는 오디오 데이터 블록리스트에서 ID를 제거하는 데 사용됩니다. 반환된 값 0은 호출이 성공했음을 나타냅니다.

API 프로토타입

```
ITMGContext ITMGAudioCtrl RemoveAudioBlackList(string openId)
```

매개변수	유형	설명
openId	String	블록리스트에서 삭제할 사용자 openid

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl ().RemoveAudioBlackList (openId);
```

음성 채팅 캡처 API

SDK 초기화 후 방에 진입해야 실시간 음성 채팅 관련 API 호출 가능합니다.
사용자가 UI에서 마이크 또는 스피커 활성화/비활성화 버튼을 클릭할 때 EnableMic 또는 EnableSpeaker API를 호출하는 것이 좋습니다.
사용자가 UI에서 마이크 버튼을 눌렀다가 놓아 말을 멈출 수 있도록 하려면 방 입장 중에 EnableAudioCaptureDevice를 한 번 호출하고 EnableAudioSend를 호출하여 사용자가 버튼을 누른 상태에서 말할 수 있도록 하는 것이 좋습니다.

API	API 설명
EnableMic	마이크 활성화/비활성화
GetMicState	마이크 상태 가져오기
EnableAudioCaptureDevice	캡처 장치 활성화/비활성화
IsAudioCaptureDeviceEnabled	캡처 장치 상태 가져오기
EnableAudioSend	오디오 업스트림 활성화/비활성화
IsAudioSendEnabled	오디오 업스트림 상태 가져오기
GetMicLevel	실시간 마이크 볼륨 레벨 가져오기
GetSendStreamLevel	실시간 오디오 업스트림 볼륨 가져오기
SetMicVolume	마이크 볼륨 설정

GetMicVolume	마이크 볼륨 레벨 가져오기

마이크 활성화 또는 비활성화

이 API는 마이크를 활성화/비활성화하는 데 사용됩니다. 방 입장 후 마이크와 스피커는 기본적으로 활성화되지 않습니다. **EnableMic = EnableAudioCaptureDevice + EnableAudioSend**

API 프로토타입

```
ITMGAudioCtrl EnableMic(bool isEnabled)
```

매개변수	유형	설명
isEnabled	boolean	마이크를 활성화하려면 이 매개변수를 true로 설정합니다. 그렇지 않으면 false로 설정합니다.

예시 코드

```
//마이크 활성화
ITMGContext.GetInstance().GetAudioCtrl().EnableMic(true);
```

마이크 상태 가져오기

이 API는 마이크 상태를 가져오는 데 사용됩니다. 반환 값 0은 마이크가 꺼져 있음을 나타내고 1은 켜져 있음을 나타냅니다.

API 프로토타입

```
ITMGAudioCtrl GetMicState()
```

예시 코드

```
micToggle.isOn = ITMGContext.GetInstance().GetAudioCtrl().GetMicState();
```

캡처 장치 활성화 또는 비활성화

이 API는 캡처 장치를 활성화/비활성화하는 데 사용됩니다. 방 입장 후 장치는 기본적으로 활성화되지 않습니다.
이 API는 방 입장 후에만 호출할 수 있습니다. 방에서 나가면 장치가 자동으로 비활성화됩니다.
권한 적용 및 볼륨 유형 조정과 같은 작업은 일반적으로 모바일 장치에서 캡처 장치가 활성화된 경우에 수행됩니다.

API 프로토타입

```
ITMGAudioCtrl int EnableAudioCaptureDevice(bool isEnabled)
```

매개변수	유형	설명
isEnabled	bool	캡처 장치를 활성화하려면 이 매개변수를 true로 설정합니다. 그렇지 않으면 false로 설정합니다.

예시 코드

```
//캡처 장치 활성화
ITMGContext.GetInstance().GetAudioCtrl().EnableAudioCaptureDevice(true);
```

캡처 장치 상태 가져오기

이 API는 캡처 장치의 상태를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGAudioCtrl bool IsAudioCaptureDeviceEnabled()
```

예시 코드

```
bool IsAudioCaptureDevice = ITMGContext.GetInstance().GetAudioCtrl().IsAudioCapture
```

오디오 스트림 전송 활성화/비활성화

이 API는 오디오 스트림 전송을 활성화/비활성화하는 데 사용됩니다. 캡처 장치가 이미 활성화된 경우 캡처된 오디오 데이터를 보냅니다. 그렇지 않으면 음소거 상태로 유지됩니다. 캡처 장치를 활성화/비활성화하는 방법에 대한 자세한 내용은 EnableAudioCaptureDevice API를 참고하십시오.

API 프로토타입

```
ITMGAudioCtrl int EnableAudioSend(bool isEnabled)
```

매개변수	유형	설명
isEnabled	bool	오디오 업스트림을 활성화하려면 이 매개변수를 true로 설정합니다. 그렇지 않으면 false로 설정합니다.

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().EnableAudioSend(true);
```

오디오 스트림 전송 상태 가져오기

이 API는 오디오 스트림 전송 상태를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGAudioCtrl bool IsAudioSendEnabled()
```

예시 코드

```
bool IsAudioSend = ITMGContext.GetInstance().GetAudioCtrl().IsAudioSendEnabled();
```

실시간 마이크 볼륨 가져오기

이 API는 실시간 마이크 볼륨을 가져오는 데 사용됩니다. 0 - 100 범위의 int 유형 값이 반환됩니다. 이 API는 20ms마다 한 번씩 호출하는 것이 좋습니다.

API 프로토타입

```
ITMGAudioCtrl int GetMicLevel
```

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetMicLevel();
```

실시간 오디오 스트림 전송 볼륨 가져오기

이 API는 로컬 실시간 오디오 스트림 전송 볼륨을 가져오는 데 사용됩니다. 0 - 100 범위의 int 유형 값이 반환됩니다.

API 프로토타입

```
ITMGAudioCtrl int GetSendStreamLevel()
```

예시 코드

```
int Level = ITMGContext.GetInstance().GetAudioCtrl().GetSendStreamLevel();
```

마이크 소프트웨어 볼륨 설정

이 API는 마이크 볼륨 레벨을 설정하는 데 사용됩니다. 해당 매개변수는 volume이며, 이는 캡처된 사운드에 대한 감쇠 또는 이득과 같습니다.

API 프로토타입

```
ITMGAudioCtrl SetMicVolume(int volume)
```

매개변수	유형	설명
volume	int	값 범위: 0-200. 기본값: 100. 0은 오디오가 음소거됨을 나타내고 100은 볼륨 레벨이 변경되지 않음을 나타냅니다.

예시 코드

```
int micVol = (int)(value * 100);
ITMGContext.GetInstance().GetAudioCtrl().SetMicVolume (micVol);
```

마이크 소프트웨어 볼륨 가져오기

이 API는 마이크 볼륨을 가져오는 데 사용됩니다. int 값이 반환됩니다. 값 101은 API SetMicVolume이 호출되지 않았음을 나타냅니다.

API 프로토타입

```
ITMGAudioCtrl GetMicVolume()
```

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetMicVolume();
```

음성 채팅 재생 API

API	API 설명
EnableSpeaker	스피커 활성화/비활성화
GetSpeakerState	스피커 상태 가져오기
EnableAudioPlayDevice	재생 장치 활성화/비활성화

IsAudioPlayDeviceEnabled	재생 장치 상태 가져오기
EnableAudioRecv	오디오 다운스트림 활성화/비활성화
IsAudioRecvEnabled	오디오 다운스트림 상태 가져오기
GetSpeakerLevel	실시간 스피커 볼륨 레벨 가져오기
GetRecvStreamLevel	방에 있는 다른 구성원의 실시간 다운스트림 오디오 레벨 가져오기
SetSpeakerVolume	스피커 볼륨 레벨 설정
GetSpeakerVolume	스피커 볼륨 레벨 가져오기

스피커 활성화 또는 비활성화

이 API는 스피커를 활성화/비활성화하는 데 사용됩니다. **EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv**

API 프로토타입

```
ITMGAudioCtrl EnableSpeaker(bool isEnabled)
```

매개변수	유형	설명
isEnabled	bool	스피커를 비활성화하려면 이 매개변수를 false로 설정합니다. 그렇지 않으면 true로 설정합니다.

예시 코드

```
//스피커 활성화
ITMGContext.GetInstance().GetAudioCtrl().EnableSpeaker(true);
```

스피커 상태 가져오기

이 API는 스피커 상태를 가져오는 데 사용됩니다. 0은 스피커가 꺼져 있음을 나타내고 1은 켜져 있음을 나타냅니다.

API 프로토타입

```
ITMGAudioCtrl GetSpeakerState()
```

예시 코드

```
speakerToggle.isOn = ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerState();
```

재생 장치 활성화 또는 비활성화

이 API는 재생 장치를 활성화/비활성화하는 데 사용됩니다.

API 프로토타입

```
ITMGAudioCtrl EnableAudioPlayDevice(bool isEnabled)
```

매개변수	유형	설명
isEnabled	bool	재생 장치를 비활성화하려면 이 매개변수를 false로 설정합니다. 그렇지 않으면 true로 설정합니다.

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().EnableAudioPlayDevice(true);
```

재생 장치 상태 가져오기

이 API는 재생 장치의 상태를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGAudioCtrl bool IsAudioPlayDeviceEnabled()
```

예시 코드

```
bool IsAudioPlayDevice = ITMGContext.GetInstance().GetAudioCtrl().IsAudioPlayDevice
```

오디오 스트림 수신 활성화/비활성화

이 API는 오디오 스트림 수신을 활성화/비활성화하는 데 사용됩니다. 재생 장치가 이미 활성화된 경우 방에 있는 다른 구성원의 오디오 데이터를 재생합니다. 그렇지 않으면 음소거 상태로 유지됩니다. 재생 장치를 활성화/비활성화하는 방법에 대한 자세한 내용은 EnableAudioPlayDevice API를 참고하십시오.

API 프로토타입

```
ITMGAudioCtrl int EnableAudioRecv(bool isEnabled)
```

매개변수	유형	설명
isEnabled	bool	오디오 다운스트림을 활성화하려면 이 매개변수를 true로 설정합니다. 그렇지 않으면 false로 설정합니다.

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().EnableAudioRecv(true);
```

오디오 스트림 수신 상태 가져오기

이 API는 오디오 스트림 수신 상태를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGAudioCtrl bool IsAudioRecvEnabled()
```

예시 코드

```
bool IsAudioRecv = ITMGContext.GetInstance().GetAudioCtrl().IsAudioRecvEnabled();
```

실시간 스피커 볼륨 가져오기

이 API는 실시간 스피커 볼륨을 가져오는 데 사용됩니다. 볼륨을 나타내기 위해 int 유형 값이 반환됩니다. 이 API는 20ms마다 한 번씩 호출하는 것이 좋습니다.

API 프로토타입

```
ITMGAudioCtrl GetSpeakerLevel()
```

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerLevel();
```

방에 있는 다른 구성원의 오디오 스트림 볼륨 가져오기

이 API는 수신된 방에 있는 다른 구성원의 실시간 오디오 스트림 볼륨을 가져오는 데 사용됩니다. int 유형 값이 반환됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGAudioCtrl int GetRecvStreamLevel(string openId)
```

매개변수	유형	설명
openId	string	방에 있는 다른 구성원의 openId

예시 코드

```
int Level = ITMGContext.GetInstance().GetAudioCtrl().GetRecvStreamLevel(openId);
```

방 구성원의 볼륨을 동적으로 설정

이 API는 방 구성원의 말하기 볼륨을 설정하는 데 사용되며 이 설정은 로컬에서만 적용됩니다.

API 프로토타입

```
public abstract int SetSpeakerVolumeByOpenID(string openid, int volume);
```

매개변수	유형	설명
openId	String	볼륨 조절이 필요한 OpenID
volume	int	백분율, [0-200] 권장, 기본값은 100

스피커 볼륨 설정

이 API는 스피커 볼륨을 설정하는 데 사용됩니다.

API 프로토타입

```
ITMGAudioCtrl SetSpeakerVolume(int volume)
```

매개변수	유형	설명
volume	int	값 범위: 0 – 200. 기본값: 100. 0은 오디오가 음소거됨을 나타내고 100은 볼륨 레벨이 변경되지 않음을 나타냅니다.

예시 코드

```
int speVol = (int)(value * 100);
ITMGContext.GetInstance().GetAudioCtrl().SetSpeakerVolume(speVol);
```

스피커 볼륨 가져오기

이 API는 스피커 볼륨을 가져오는 데 사용됩니다. 볼륨을 나타내기 위해 int 유형 값이 반환됩니다. 101은 SetSpeakerVolume API가 호출되지 않았음을 나타냅니다.
Level은 실시간 Volume을 나타내고 Volume 은 스피커 볼륨을 나타냅니다. 최종 볼륨 = Level × Volume %. 예를 들어 레벨이 100이고 Volume이 60이면 최종 볼륨은 60입니다.

API 프로토타입

```
ITMGAudioCtrl GetSpeakerVolume()
```

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerVolume();
```

장치 선택 API

장치 선택 API는 PC에서만 사용할 수 있습니다.

API	API 설명
GetMicListCount	마이크 수 가져오기
GetMicList	마이크 열거
GetSpeakerListCount	획득한 스피커 장치 수
GetSpeakerList	스피커 열거
SelectMic	마이크 선택
SelectSpeaker	스피커 선택

마이크 수 가져오기

이 API는 마이크 수를 가져오는 데 사용됩니다.

함수 프로토타입

```
public abstract int GetMicListCount()
```

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetMicListCount();
```

마이크 열거

이 API는 마이크를 열거하기 위해 GetMicListCount API와 함께 사용됩니다.

함수 프로토타입

```
public abstract int GetMicList(out List<TMGAudioDeviceInfo> devicesInfo, int count)
```

매개변수	유형	설명
ppDeviceInfoList	TMGAudioDeviceInfo	장치 목록
count	int	획득한 마이크 장치 수

TMGAudioDeviceInfo 매개변수	유형	설명
m_strDeviceID	string	장치 이름
m_strDeviceID	string	장치 ID

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetMicList(devicesInfo,count);
```

마이크 선택

이 API는 마이크를 선택하는 데 사용됩니다. 이 API가 호출되지 않거나 DEVICEID_DEFAULT가 전달되면 기본 마이크가 선택됩니다.
GetMicList API에서 반환되는 0번째 장치 id는 통화 장치의 기본 장치입니다. 선택된 통화 장치가 있으면 서비스에서 유지합니다. 만약 통화 장치의 플러그가 뽑히면 통화 장치가 다시 기본 장치로 변경됩니다.

함수 프로토타입

```
public abstract int SelectMic(string micID);
```

매개변수	유형	설명
pMicID	string	GetMicList에서 반환된 목록의 마이크 ID입니다.

예시 코드

```
string deviceID = DEVICE_ID_DEFAULT;
                if (index != 0)
                {
                    deviceID = listMicInfo[index - 1].m_strDeviceID;
                }
                ITMGContext.GetInstance().GetAudioCtrl().SelectMic(deviceID);
                selectedMicID = deviceID;
```

이 API는 스피커 수를 가져오는 데 사용됩니다.

함수 프로토타입

```
public abstract int GetSpeakerListCount();
```

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerListCount();
```

스피커 열거

이 API는 GetSpeakerListCount API와 함께 스피커를 열거하는 데 사용됩니다.

함수 프로토타입

```
public abstract int GetSpeakerList(out List<TMGAudioDeviceInfo> devicesInfo, int co
```

매개변수	유형	설명
ppDeviceInfoList	TMGAudioDeviceInfo	장치 목록
count	int	획득한 스피커 장치 수

TMGAudioDeviceInfo 매개변수	유형	설명
m_strDeviceID	string	장치 이름
m_strDeviceID	string	장치 ID

예시 코드

```
int speakerCount = ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerListCount();
Debug.LogFormat("speakerCount = {0}", speakerCount);
if (speakerCount > 0)
    {
        int ret = ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerList(out listS
        Debug.LogFormat("GetSpeakerList ret = {0}", ret);
        if (ret != 0)
        {
            listSpeakerInfo = null;
        }
    }
}
```

스피커 선택

이 API는 재생 장치를 선택하는 데 사용됩니다. 이 API가 호출되지 않거나 "DEVICEID_DEFAULT"가 전달되면 기본 재생 장치가 선택됩니다.

함수 프로토타입

```
public abstract int SelectSpeaker(string speaker);
```

매개변수	유형	설명
speaker	string	GetSpeakerList에 의해 반환된 목록의 스피커 ID입니다.

예시 코드

```
speakerDropdown = transform.Find("DevicePanel/SpeakerSelect").GetComponent<Dropdown
        if (speakerDropdown != null)
        {
            speakerDropdown.onValueChanged.AddListener(delegate (int index)
            {
                string deviceID = DEVICE_ID_DEFAULT;
                if (index != 0)
                {
                    deviceID = listSpeakerInfo[index - 1].m_strDeviceID;
                }
                ITMGContext.GetInstance().GetAudioCtrl().SelectSpeaker(deviceID);
                selectedSpeakerID = deviceID;
            });
        }
```


특수 API

인이어 모니터링 활성화

이 API는 인이어 모니터링을 활성화하는 데 사용됩니다. 자신의 목소리를 듣기 전에 EnableLoopBack+EnableSpeaker를 호출해야 합니다.

API 프로토타입

```
ITMGContext GetAudioCtrl EnableLoopBack(bool enable)
```

매개변수	유형	설명
enable	bool	사용 여부 지정

예시 코드

```
ITMGContext.GetInstance().GetAudioCtrl().EnableLoopBack(true);
```

장치 사용 및 릴리스에 대한 콜백

방에서 장치를 사용하거나 해제한 후에 이벤트 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.

```
public delegate void QAVOnDeviceStateChangedEvent(int deviceType, string deviceId,
public abstract event QAVOnDeviceStateChangedEvent OnDeviceStateChangedEvent;
```

매개변수	유형	설명
deviceType	int	1. 캡처 장치 2. 재생 장치
deviceId	string	장치 GUID. 장치를 식별하는 데 사용되며 Windows 및 Mac에만 적용됩니다.
openOrClose	bool	캡처/재생 장치 사용 또는 릴리스

예시 코드

```
이벤트 수신:
ITMGContext.GetInstance().GetAudioCtrl().OnDeviceStateChangedEvent += new QAVAudioD
수신된 이벤트 처리:
void QAVAudioDeviceStateCallback(int deviceType, string deviceId, bool openOrClose)
    //장치 점유 및 릴리스에 대한 콜백
}
```

사용자 방 오디오 유형 가져오기

이 API는 사용자의 방 오디오 유형을 가져오는 데 사용됩니다. 반환되는 값은 방 오디오 유형입니다. 값 0은 사용자의 방 오디오 유형을 가져오는 동안 오류가 발생했음을 나타냅니다. 방 오디오 유형에 대해서는 EnterRoom API를 참고하십시오.

API 프로토타입

```
ITMGContext ITMGRoom public  int GetRoomType()
```

예시 코드

```
ITMGContext.GetInstance().GetRoom().GetRoomType();
```

방 유형 변경

이 API는 사용자의 방 오디오 유형을 수정하는 데 사용됩니다. 결과는 콜백 이벤트를 참고하십시오. 이벤트 유형은 ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE입니다. 방의 오디오 유형은 방에 들어오는 첫 번째 사용자에 의해 결정됩니다. 이후 방에 있는 구성원이 방 유형을 변경하면 모든 구성원에게 적용됩니다.

API 프로토타입

```
ITMGContext ITMGRoom public int ChangeRoomType(ITMGRoomType roomtype)
```

매개변수	유형	설명
roomtype	ITMGRoomType	전환할 방 유형입니다. 방 오디오 유형에 대해서는 EnterRoom API를 참고하십시오.

예시 코드

```
ITMGContext.GetInstance().GetRoom().ChangeRoomType(ITMG_ROOM_TYPE_FLUENCY);
```

콜백 이벤트

방 유형을 설정합니다. 방 유형이 설정되면 델리게이트 함수를 통해 콜백을 실행하여 수정이 완료되었음을 알리는 메시지를 전달합니다.

반환된 매개변수	설명
roomtype	전환된 roomtype 반환

```
public abstract event QAVCallback OnChangeRoomtypeCallback;
public abstract event QAVOnRoomTypeChangedEvent OnRoomTypeChangedEvent;
```

예시 코드

```
//이벤트 수신:
ITMGContext.GetInstance ().OnRoomTypeChangedEvent += new QAVOnRoomTypeChangedEvent
//수신한 이벤트 처리:
void OnRoomTypeChangedEvent(int roomtype)
{
        ShowWarnning (string.Format ("RoomTypeChanged current:{0}",roomtype));
}
```

방 유형 변경 알림

방 유형이 귀하 또는 방의 다른 사용자에 의해 변경되면 이 알림 이벤트는 비즈니스 레이어에 방 유형 변경을 알리는 데 사용됩니다. 반환되는 값은 방 유형입니다. 자세한 내용은 EnterRoom API를 참고하십시오.

```
public delegate void QAVOnRoomTypeChangedEvent(int roomtype);
public abstract event QAVOnRoomTypeChangedEvent OnRoomTypeChangedEvent;
```

예시 코드

```
//이벤트 수신:
ITMGContext.GetInstance().OnRoomTypeChangedEvent += new QAVOnRoomTypeChangedEvent(O
//수신한 이벤트 처리:
void OnRoomTypeChangedEvent(int roomtype){
    //방 유형 변경 후 콜백 전송
}
```

방 통화 품질 모니터링 이벤트

네트워크 품질을 수신하는 데 사용되는 품질 모니터링 이벤트입니다. 네트워크 상태가 좋지 않으면 비즈니스 레이어에서 UI를 통해 네트워크를 전환하도록 요청합니다. 이 이벤트는 방 입장 후 2초마다 한 번씩 발생하며 메시지는 ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY입니다. 반환된 매개변수에는 weight, loss 및 delay가 포함되며 아래에 자세히 설명되어 있습니다.

매개변수	유형	설명
weight	int	값 범위: 1 - 50. 50은 우수한 음질을 나타내고, 1은 매우 열악한(거의 사용할 수 없는) 음질을 나타내며, 0은 초기 의미 없는 값을 나타냅니다. 일반적으로 값이 30 미만이

		면 비즈니스 레이어에서 사용자에게 네트워크 상태가 좋지 않음을 알리고 네트워크 전환을 권장합니다.
loss	double	업스트림 패킷 손실률.
delay	int	음성 채팅 지연(ms).

버전 번호 가져오기

이 API는 분석을 위한 SDK 버전 번호를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGContext  abstract string GetSDKVersion()
```

예시 코드

```
ITMGContext.GetInstance().GetSDKVersion();
```

로그 출력 레벨 설정

이 API는 출력할 로그의 수준을 설정하는 데 사용되며 초기화 전에 호출해야 합니다. 기본 수준을 유지하는 것이 좋습니다.

API 프로토타입

```
ITMGContext  SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

매개변수 설명

매개변수	유형	설명
levelWrite	ITMG_LOG_LEVEL	기록할 로그 수준을 설정합니다. TMG_LOG_LEVEL_NONE은 쓰지 않음을 나타냅니다. 기본값: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	출력할 로그의 수준을 설정합니다. TMG_LOG_LEVEL_NONE은 출력하지 않음을 나타냅니다. 기본값: TMG_LOG_LEVEL_ERROR

ITMG_LOG_LEVEL은 아래와 같습니다.

ITMG_LOG_LEVEL	설명
TMG_LOG_LEVEL_NONE	로그를 출력하지 않음
TMG_LOG_LEVEL_ERROR	오류 로그 출력(기본값)

TMG_LOG_LEVEL_INFO	정보 로그 출력
TMG_LOG_LEVEL_DEBUG	디버깅 로그 출력
TMG_LOG_LEVEL_VERBOSE	고빈도 로그 출력

예시 코드

```
ITMGContext.GetInstance().SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

로그 출력 경로 설정

이 API는 로그 출력 경로를 설정하는 데 사용됩니다. 기본 경로는 다음과 같습니다. Init 전에 호출해야 합니다.

플랫폼	경로
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API 프로토타입

```
ITMGContext   SetLogPath(string logDir)
```

매개변수	유형	설명
logDir	String	경로

예시 코드

```
ITMGContext.GetInstance().SetLogPath(path);
```

진단 메시지 가져오기

이 API는 실시간 음성/화상 통화 품질에 대한 정보를 얻는 데 사용되며, 주로 실시간 통화 품질을 보고 문제를 해결하는 데 사용되며 비즈니스 측면에서는 무시할 수 있습니다.

API 프로토타입


```
ITMGRoom GetQualityTips()
```

예시 코드

```
string tips = ITMGContext.GetInstance().GetRoom().GetQualityTips();
```


음성-텍스트 변환 서비스

최종 업데이트 날짜: : 2024-01-18 16:05:38

본문은 Unity 음성 메시지 및 음성-텍스트 변환 서비스용 GME(Game Multimedia Engine) 클라이언트 API를 통합하고 디버깅하는 방법을 설명합니다.

GME 사용을 위한 주요 고려 사항

GME는 Init 및 Poll과 같은 핵심 API에 의존하는 실시간 음성, 음성 메시지 및 음성 텍스트 변환 서비스를 제공합니다.

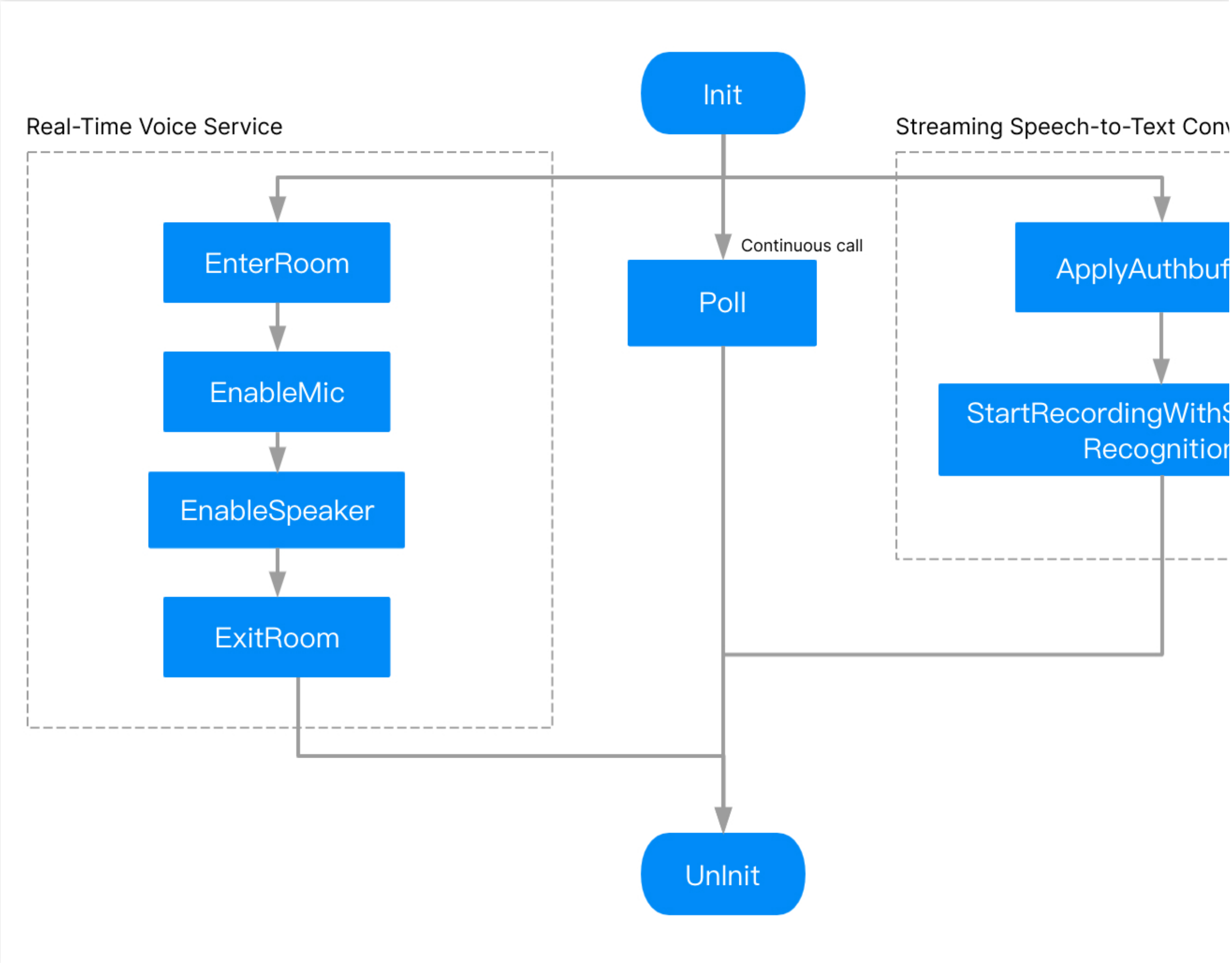
주요 사항

서비스 활성화에 설명된 대로 GME 애플리케이션을 생성하고 SDK의 AppID 및 Key를 가져옵니다.
GME 실시간 음성 채팅 서비스, 음성 메시지 서비스 및 텍스트 변환 서비스를 활성화합니다. 자세한 내용은 서비스 활성화 가이드를 참고하십시오.
GME를 사용하기 전에 프로젝트를 구성하십시오. 그렇지 않으면 SDK가 적용되지 않습니다.
GME API가 성공적으로 호출되면 QAVError.OK가 0 값으로 반환됩니다.
GME API는 동일한 스레드에서 호출되어야 합니다.
GME가 이벤트 콜백을 트리거하려면 Poll API를 주기적으로 호출해야 합니다.
에러 코드 세부 정보는 에러 코드를 참고하십시오.
주의사항 :
음성-텍스트 변환 API에는 기본 빈도 제한이 있습니다. 한도 내 호출 과금 방식에 대한 자세한 정보는 구매 가이드를 참고하십시오. 한도를 늘리거나 한도 초과 시 호출 요금의 과금 방식 문의는 영업팀에 연락하거나 티켓 제출 하십시오.
음성 메시지 비스트리밍 텍스트 변환 API **SpeechToText()**: 단일 계정 기본 동시 연결 수는 10개 입니다
음성 메시지 스트리밍 텍스트 변환 API **StartRecordingWithStreamingRecognition()**: 단일 계정 기본 동시 연결 수는 50개 입니다
실시간 음성 채팅 스트리밍 텍스트 변환 API **StartRealTimeASR()**: 단일 계정 기본 동시 연결 수는 50개 입니다

SDK에 연결

중요 단계

SDK 연결 관련 주요 프로세스는 다음과 같습니다.



1. GME 초기화
2. 주기적 Poll 호출을 통해 콜백 트리거
3. 인증 초기화
4. 스트리밍 음성 인식 시작
5. 녹음 중지
6. GME 초기화 취소

C# 클래스

클래스	설명
ITMGContext	주요 API
ITMGPTT	음성 메시지 및 음성-텍스트 변환 API

주요 API

API	API 설명
Init	GME 초기화
Poll	이벤트 콜백 트리거
Pause	시스템 일시 중지
Resume	시스템 복구
Uninit	GME 초기화 취소

헤더 파일 가져오기

```
using GME;
```

인스턴스 가져오기

QAVContext.GetInstance() 대신 ITMGContext 메소드를 사용하여 Context 인스턴스를 가져옵니다.

SDK 초기화

실시간 음성, 음성 메시지, 음성 텍스트 변환 서비스를 사용하려면 먼저 **Init API를 통해 SDK를 초기화해야 합니다**. Init API는 다른 API와 동일한 스레드에서 호출해야 합니다. 기본 스레드에서 모든 API를 호출하는 것이 좋습니다.

API 프로토타입

```
//class ITMGContext
public abstract int Init(string sdkAppID, string openID);
```

매개변수	유형	설명
sdkAppId	string	GME 콘솔에서 제공되는 AppID로, 서비스 활성화의 안내에 따라 얻을 수 있습니다.
openID	string	openID는 Int64 유형만 가능하며 string으로 변환되어 전달됩니다. 해당 규칙을 사용자 정의할 수 있으며 App에서 고유해야 합니다. Openid를 문자열로 전달하려면 Submit Ticket하여 신청하십시오.

반환된 값

반환 값	처리
QAVError.OK= 0	SDK 초기화 성공
AV_ERR_SDK_NOT_FULL_UPDATE=7015	SDK 파일이 완전한지 확인합니다. 삭제한 후 SDK를 다시 가져

	오는 것이 좋습니다

7015 오류 메시지

7015 에러 코드는 md5로 판단됩니다. 통합 중에 이 오류가 보고되면 메시지에 따라 SDK 파일의 무결성과 버전을 확인하십시오.

반환 값 AV_ERR_SDK_NOT_FULL_UPDATE는 사전 **알림일 뿐**이며 초기화 실패를 일으키지는 않습니다.

타사 강화, Unity 패키징 메커니즘 및 기타 요인으로 인해 라이브러리 파일의 md5가 영향을 받아 오판이 발생할 수 있습니다. **정식 출시를 위한 로직에서는 이 오류를 무시**하고 UI에 표시하지 않도록 하십시오.

예시 코드

```
int ret = ITMGContext.GetInstance().Init(sdkAppId, openID);
// 반환된 값으로 초기화 성공 여부 판단
if (ret != QAVError.OK)
    {
        Debug.Log("SDK 초기화 실패:"+ret);
        return;
    }
```

이벤트 콜백 트리거

이벤트 콜백은 update에서 Poll API를 주기적으로 호출하여 트리거할 수 있습니다. Poll API는 GME의 메시지 펌프이며 GME가 이벤트 콜백을 트리거하도록 주기적으로 호출해야 합니다. 그렇지 않으면 전체 SDK 서비스가 비정상적으로 실행됩니다. 자세한 내용은 SDK 다운로드 가이드의 EnginePollHelper 파일을 참고하십시오.

주의사항：

비정상적인 API 콜백을 방지하기 위해 Poll API는 주기적으로 메인 스레드에서 호출되어야 합니다.

API 프로토타입

```
ITMGContext public abstract int Poll();
```

예시 코드

```
public void Update()
    {
        ITMGContext.GetInstance().Poll();
    }
```

시스템 일시 중지

시스템에서 Pause 이벤트가 발생하면 엔진에도 일시 중지를 알려야 합니다. 예를 들어 애플리케이션이 백그라운드로 전환되고(OnApplicationPause, isPause=True) 방에서 오디오를 재생하기 위해 백그라운드가 필요하지 않은 경우

Pause API를 호출하여 GME 서비스를 일시 중지하십시오.

API 프로토타입

```
ITMGContext public abstract int Pause()
```

시스템 복구

시스템에서 Resume 이벤트가 발생하면 엔진에도 Resume에 대해 알려야 합니다. Resume API는 음성 채팅 복구만 지원합니다.

API 프로토타입

```
ITMGContext  public abstract int Resume()
```

SDK 초기화 취소

이 API는 SDK 초기화 취소를 통해 초기화를 해제하는 데 사용됩니다. **게임 비즈니스 계정이 openid에 바인딩되어 있는 경우 게임 계정을 전환하려면 GME를 초기화 취소한 다음 새 openid를 사용하여 다시 초기화해야 합니다.**

API 프로토타입

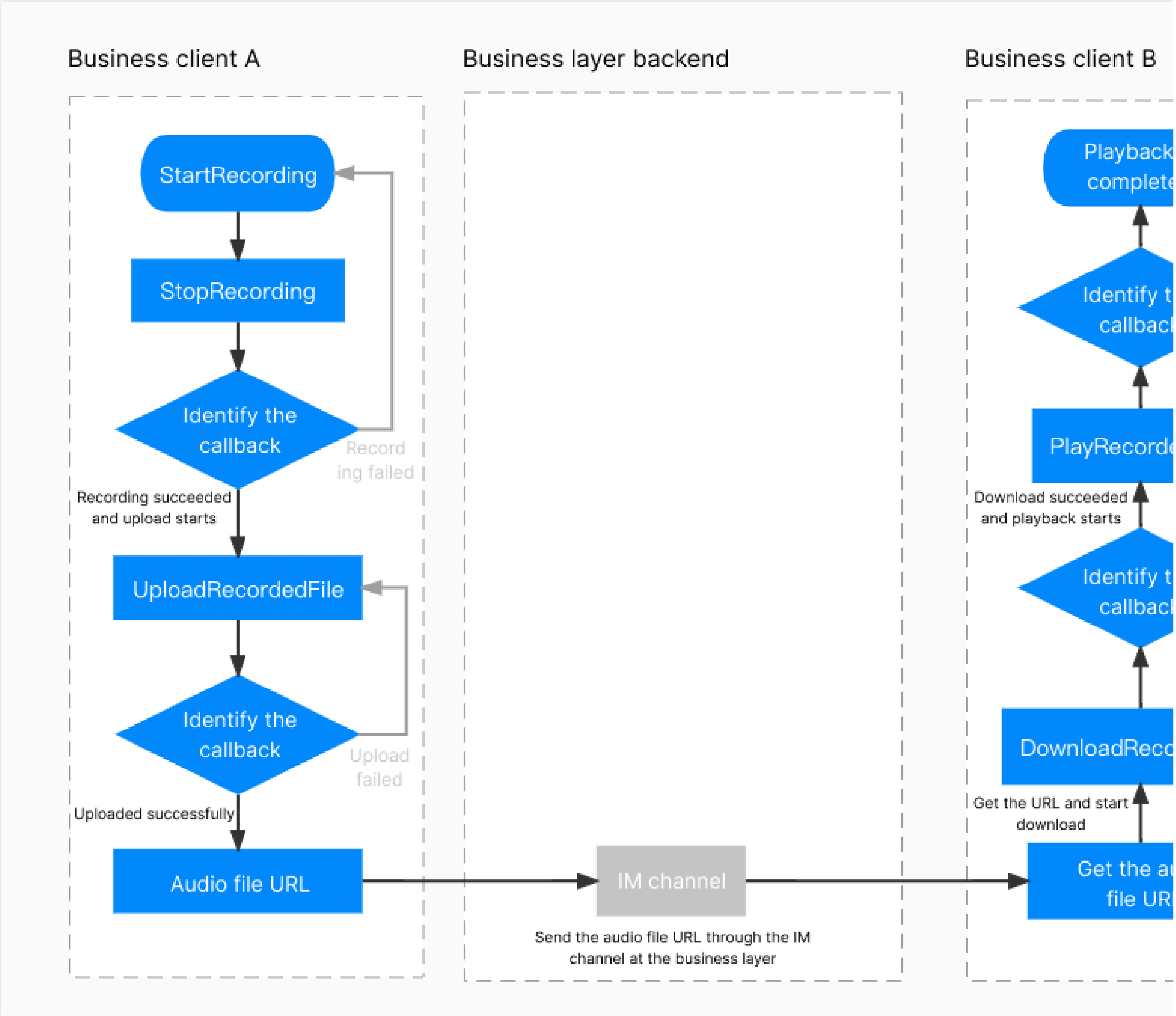
```
ITMGContext public abstract int Uninit()
```

음성 메시지 서비스 및 음성-텍스트 변환 서비스

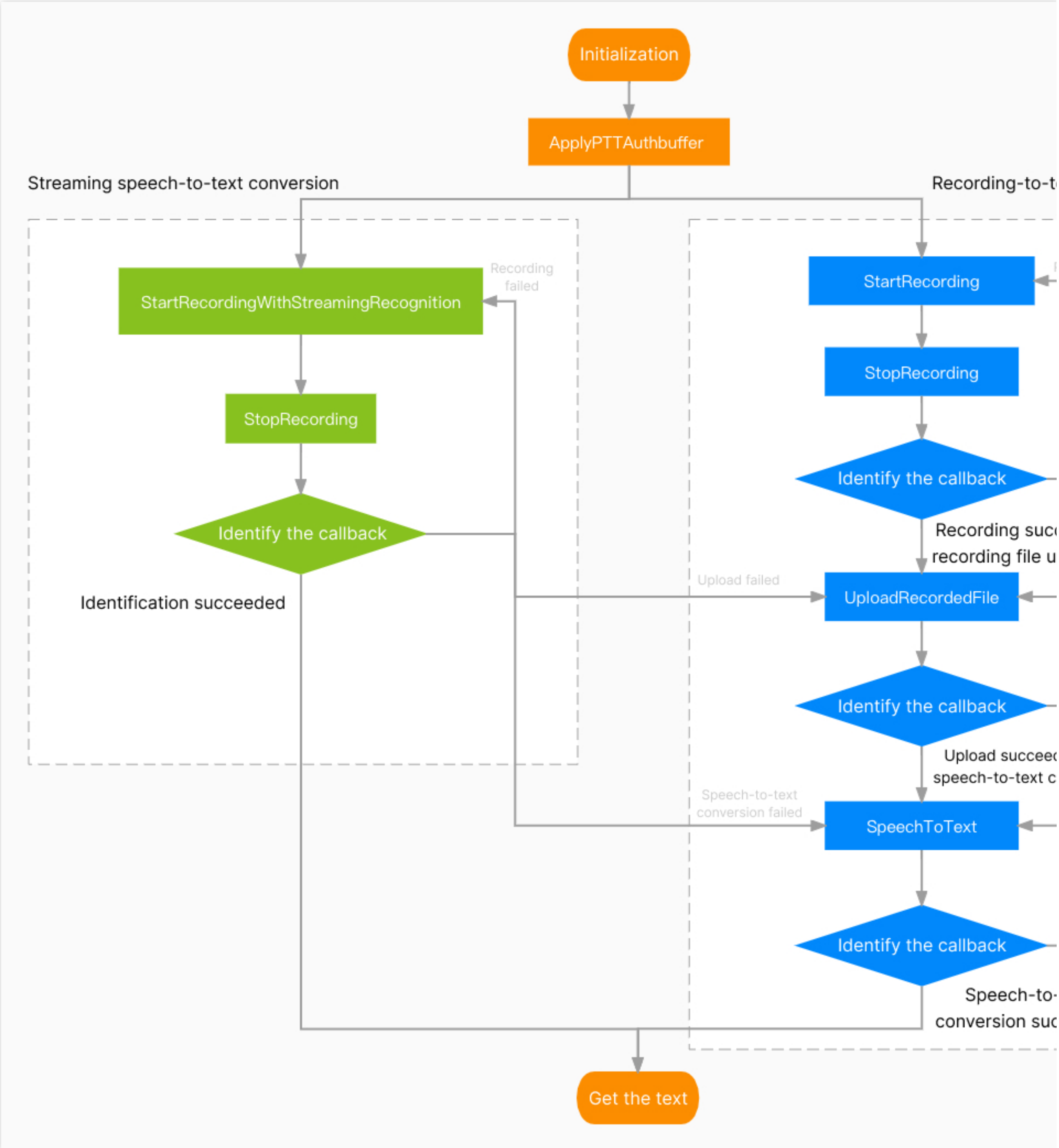
설명：

음성-텍스트 변환 서비스는 빠른 녹음 파일 텍스트 변환 및 음성 메시지 스트리밍 텍스트 변환으로 구성됩니다.
음성 메시지 서비스 이용 시 음성 채팅방에 입장할 필요가 없습니다.
음성 메시지의 기본 녹음 시간은 1 - 58초입니다. 최대 녹음 시간을 10초로 수정하는 등 녹음 시간을 사용자 지정하려면 초기화 후 SetMaxMessageLength API를 호출하여 설정합니다.

음성 메시지 서비스 사용


Business client A
Business layer backend
Business client B
StartRecording
StopRecording
Identify the callback
Record ing failed
Recording succeeded and upload starts
UploadRecordedFile
Identify the callback
Upload failed
Uploaded successfully
Audio file URL
IM channel
Send the audio file URL through the IM channel at the business layer
Download succeeded and playback starts
Get the URL and start download

텍스트 변환 서비스 사용



API	API 설명
GenAuthBuffer	로컬 인증 키 생성
ApplyPTTAuthbuffer	인증 초기화
SetMaxMessageLength	음성 메시지의 최대 길이 지정

로컬 인증 키 생성

관련 기능의 암호화 및 인증을 위해 AuthBuffer를 생성합니다. 프로덕션 환경에서 릴리스하려면 인증 키에 설명된 대로 백엔드 배포 키를 사용하십시오.

API 프로토타입

```
QAVAuthBuffer GenAuthBuffer(int appId, string roomId, string openId, string key)
```

매개변수	유형	설명
appId	int	Tencent Cloud 콘솔의 AppId.
roomId	string	null 또는 빈 문자열 입력.
openId	string	Init 시 OpenId와 동일한 사용자 ID.
key	string	Tencent Cloud 콘솔의 권한 키.

애플리케이션 인증

인증 정보가 생성되면 SDK에 인증이 할당됩니다.

API 프로토타입

```
ITMGPTT int ApplyPTTAuthbuffer (byte[] authBuffer)
```

매개변수	유형	설명
authBuffer	byte[]	인증

예시 코드

```
UserConfig.SetAppID(transform.Find ("appId").GetComponent<InputField> ().text);
UserConfig.SetUserID(transform.Find ("userId").GetComponent<InputField> ().text);
UserConfig.SetAuthKey(transform.Find("authKey").GetComponent<InputField>().text);
byte[] authBuffer = UserConfig.GetAuthBuffer(UserConfig.GetAppID(), UserConfig.GetU
ITMGContext.GetInstance ().GetPttCtrl ().ApplyPTTAuthbuffer(authBuffer);
```

음성 메시지의 최대 길이 지정

이 API는 음성 메시지의 최대 길이(최대 58초)를 지정하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT int SetMaxMessageLength(int msTime)
```

매개변수	유형	설명
msTime	int	오디오 길이(ms), 값 범위: 1000 < msTime < = 58000

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().SetMaxMessageLength(58000);
```

스트리밍 음성 인식

음성 메시지 및 음성-텍스트 변환 API

API	API 설명
StartRecordingWithStreamingRecognition	스트리밍 녹음 시작
StopRecording	녹음 중지

스트리밍 음성 인식 시작

이 API는 스트리밍 음성 인식을 시작하는 데 사용됩니다. 콜백에서 음성-텍스트 변환된 텍스트가 실시간으로 반환됩니다. 인식할 언어를 지정하거나 음성에서 인식된 정보를 지정된 언어로 번역하여 반환할 수 있습니다. **녹음을 중지하려면 녹음 중지를 호출합니다.**

API 프로토타입

```
ITMGPTT int StartRecordingWithStreamingRecognition(string filePath)
ITMGPTT int StartRecordingWithStreamingRecognition(string filePath, string speechLa
```

매개변수	유형	설명
filePath	String	저장된 오디오 파일의 경로
speechLanguage	String	언어 매개변수를 지정하여 해당 언어로 텍스트 변환을 진행합니다. 매개변수는 Language Parameter Reference List를 참고하십시오.
translateLanguage	String	speechLanguage와 동일한 값을 입력합니다

예시 코드

```
string recordPath = Application.persistentDataPath + string.Format("/{0}.silk", sUi
int ret = ITMGContext.GetInstance().GetPttCtrl().StartRecordingWithStreamingRecogni
```

주의사항 :

번역은 추가 요금이 발생합니다. 자세한 내용은 구매 가이드를 참고하십시오.

스트리밍 음성 인식 콜백

스트리밍 음성 인식이 시작된 후 OnStreamingSpeechComplete 또는 OnStreamingSpeechisRunning 알림에서 콜백 메시지를 수신해야 합니다. 자세한 내용은 아래와 같습니다.

`OnStreamingSpeechComplete` 은 녹음이 중지되고 인식이 완료된 후 텍스트를 반환합니다. 이는 음성 단락 이후에 인식된 텍스트를 반환하는 것과 같습니다.

`OnStreamingSpeechisRunning` 은 녹음 중 실시간으로 인식된 텍스트를 반환합니다. 이는 말하는 동안 인식된 텍스트를 반환하는 것과 같습니다.

이벤트 메시지는 실제 필요에 따라 OnEvent 알림에서 식별됩니다. 전달된 매개변수에는 다음 네 가지 메시지가 포함됩니다.

메시지 이름	설명
result	스트리밍식 음성 인식 성공 여부를 판단하는 반환 코드
text	음성-텍스트 변환에서 인식된 텍스트입니다
file_path	저장된 녹음 파일의 로컬 경로입니다
file_id	90일 동안 보관되는 녹음 파일의 백엔드 url 주소

주의사항 :

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` 메시지 수신 시 file_id는 비어 있습니다.

에러 코드

에러 코드	설명	처리 방법
32775	스트리밍 음성-텍스트 변환에 실패했지만 녹음은 성공하였습니다	UploadRecordedFile API를 호출하여 녹음 파일을 업로드한 다음 SpeechToText API를 호출하여 음성을 텍스트로 변환합니다
32777	스트리밍 음성-텍스트 변환에 실패했지만, 녹음 및 업로드는 성공하였습니다	반환된 메시지에는 업로드 성공된 백엔드 url이 포함되어 있으며, SpeechToText API를 호출하여 음성을 텍스트로 변환합니다
32786	스트리밍 음성-텍스트 변환 실패입니	스트리밍 녹화 상태입니다. 스트리밍 녹화 API의 실행 결

	다	과가 반환될 때까지 기다려주십시오.
32787	음성-텍스트 변환에 성공했지만, 텍스트 번역 서비스가 활성화되지 않았습니다	콘솔에서 텍스트 번역 서비스를 활성화해야 합니다
32788	음성-텍스트 변환에 성공했지만, 텍스트 번역 서비스의 언어 매개변수가 잘못되었습니다	전달된 매개변수를 확인합니다

오류 코드 4098이 보고되면 해결 방법은 FAQ를 참고하십시오.

예시 코드

```
            //이벤트 수신:
            ITMGContext.GetInstance().GetPttCtrl().OnStreamingSpeechComplete +=new
            ITMGContext.GetInstance().GetPttCtrl().OnStreamingSpeechisRunning += ne
            //수신한 이벤트 처리:
            void OnStreamingSpeechComplete(int code, string fileid, string filepath
                    //스트리밍 음성 인식을 위한 콜백
            }

            void OnStreamingRecisRunning(int code, string fileid, string filePath,
                    if (code == 0)
                    {
                        setBtnText(mStreamBtn, "스트리밍");
                        InputField field = transform.Find("recordFilePath").GetComp
                        field.text = filePath;

                        field = transform.Find("downloadUrl").GetComponent<InputFie
                        field.text = "Stream is Running";

                        field = transform.Find("convertTextResult").GetComponent<In
                        field.text = result;
                        showWarningText("녹음 중");
                    }
}
```

음성 메시지 녹음

녹음 프로세스는 다음과 같습니다. 녹음 시작 > 녹음 중지 > 녹음 콜백 반환 > 다음 녹음 시작.

음성 메시지 및 음성-텍스트 변환 API

API	API 설명
StartRecording	녹음 시작
PauseRecording	녹음 일시정지
ResumeRecording	녹음 재개
StopRecording	녹음 중지
CancelRecording	녹음 취소

녹음 시작

이 API는 녹음을 시작하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT int StartRecording(string fileDir)
```

매개변수	유형	설명
fileDir	string	저장된 오디오 파일의 경로

예시 코드

```
string recordPath = Application.persistentDataPath + string.Format ("/{0}.silk", sU
int ret = ITMGContext.GetInstance().GetPttCtrl().StartRecording(recordPath);
```

녹음 정지

이 API는 녹음을 중지하는 데 사용됩니다. 비동기식이며 녹음이 중지된 후 녹음 완료에 대한 콜백이 반환됩니다. 녹음 파일은 녹음이 성공한 후에만 사용할 수 있습니다.

API 프로토타입

```
ITMGPTT int StopRecording()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().StopRecording();
```

녹음 시작 콜백

녹음이 완료되면 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.
녹음을 중지하려면 StopRecording을 호출합니다. 녹음 시작 콜백은 녹음이 중지된 후에 반환됩니다.

API 프로토타입

```
public delegate void QAVRecordFileCompleteCallback(int code, string filepath);
public abstract event QAVRecordFileCompleteCallback OnRecordFileComplete;
```

매개변수	유형	설명
code	string	code 0: 녹음 완료
filepath	string	녹음 파일 저장 경로로, 액세스할 수 있어야 하며 fileid일 수 없습니다

에러 코드

에러 코드 값	원인	솔루션 제안
4097	매개변수가 비어 있습니다	코드의 API 매개변수가 올바른지 확인합니다
4098	초기화 오류입니다	장치가 사용 중인지, 권한이 정상인지, 초기화가 정상인지 확인합니다
4099	녹음이 진행 중입니다	SDK 기록 기능이 적시에 사용되는지 확인합니다
4100	오디오 데이터가 캡처되지 않았습니다	마이크가 제대로 작동하는지 확인합니다
4101	녹음 중 파일 액세스 중 오류가 발생했습니다	파일의 존재와 파일 경로의 유효성을 확인합니다
4102	마이크가 인증되지 않았습니다	SDK를 사용하기 위해서는 마이크 권한이 필요하며, 권한을 추가하려면 해당 엔진 또는 플랫폼에 대한 SDK 프로젝트 구성 문서를 참고하십시오
4103	녹음 시간이 너무 짧습니다	녹음 시간은 ms 단위이어야 합니다. 매개변수가 올바른지 확인합니다. 녹음 시간이 1,000ms 이상이어야 녹음에 성공합니다.
4104	녹음 작업이 시작되지 않았습니다	녹음 시작 API가 호출되었는지 확인합니다

예시 코드

```
//이벤트 수신
```

```
ITMGContext.GetInstance().GetPttCtrl().OnRecordFileComplete +=  new QAVRecordFileCo
//수신한 이벤트 처리
void OnRecordFileComplete(int code, string filepath){
    //녹음 시작 콜백
}
```

녹음 일시정지

이 API는 녹음을 일시 중지하는 데 사용됩니다. 녹음을 재개하려면 ResumeRecording API를 호출하십시오.

API 프로토타입

```
ITMGPTT int PauseRecording()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().PauseRecording();
```

녹음 재개

이 API는 녹음을 재개하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT int ResumeRecording()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().ResumeRecording();
```

녹음 취소

이 API는 녹음 취소에 사용됩니다. **취소 후 콜백이 없습니다**.

API 프로토타입

```
ITMGPTT int CancelRecording()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().CancelRecording();
```

음성 메시지 업로드, 다운로드 및 재생

API	API 설명
UploadRecordedFile	오디오 파일 업로드
DownloadRecordedFile	오디오 파일 다운로드
PlayRecordedFile	오디오 재생
StopPlayFile	오디오 재생 중지
GetFileSize	오디오 파일 크기
GetVoiceFileDuration	오디오 파일 길이

오디오 파일 업로드

이 API는 오디오 파일을 업로드하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT int UploadRecordedFile (string filePath)
```

매개변수	유형	설명
filePath	String	업로드된 오디오 파일의 로컬 경로

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().UploadRecordedFile(filePath);
```

오디오 파일 업로드 완료 콜백

오디오 파일 업로드가 완료되면 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.

API 프로토타입

```
public delegate void QAVUploadFileCompleteCallback(int code, string filepath, strin
public abstract event QAVUploadFileCompleteCallback OnUploadFileComplete;
```

매개변수	유형	설명
code	int	code 0: 녹음 완료

filepath	string	저장된 녹음 파일의 경로
fileid	string	파일 URL 경로

에러 코드

에러 코드 값	원인	솔루션 제안
8193	업로드 시 파일에 액세스하는 동안 오류가 발생했습니다	파일의 존재와 파일 경로의 유효성을 확인합니다
8194	서명 확인에 실패했습니다	인증키가 맞는지, 음성 메시지 및 음성 변환 기능이 초기화 되었는지 확인합니다
8195	네트워크 오류가 발생했습니다	장치가 인터넷에 액세스할 수 있는지 확인합니다
8196	업로드 매개변수를 가져오는 동안 네트워크 실패가 발생했습니다	인증이 올바른지, 장치 네트워크가 정상적으로 인터넷에 액세스할 수 있는지 확인합니다
8197	업로드 매개변수를 가져오는 과정에서 반환된 패킷이 비어 있습니다	인증이 올바른지, 장치 네트워크가 정상적으로 인터넷에 액세스할 수 있는지 확인합니다
8198	업로드 매개변수를 가져오는 과정에서 반환된 패킷을 디코딩하지 못했습니다	인증이 올바른지, 장치 네트워크가 정상적으로 인터넷에 액세스할 수 있는지 확인합니다
8200	appinfo가 설정되지 않았습니다	apply API가 호출되었는지 또는 입력 매개변수가 비어 있는지 확인합니다

예시 코드

```
//이벤트 수신
ITMGContext.GetInstance().GetPttCtrl().OnUploadFileComplete +=new QAVUploadFileComp
//수신한 이벤트 처리
void OnUploadFileComplete(int code, string filepath, string fileid){
    //오디오 파일 업로드 완료 콜백
}
```

오디오 파일 다운로드

이 API는 오디오 파일을 다운로드하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT DownloadRecordedFile (string fileID, string downloadFilePath)
```

매개변수	유형	설명
fileID	String	파일 url
downloadFilePath	String	파일 로컬 저장 경로로, 액세스할 수 있어야 하며 fileid일 수 없습니다

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().DownloadRecordedFile(fileId, filePath);
```

오디오 파일 다운로드 완료 콜백

오디오 파일 다운로드가 완료되면 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.

API 프로토타입

```
public delegate void QAVDownloadFileCompleteCallback(int code, string filepath, str
public abstract event QAVDownloadFileCompleteCallback OnDownloadFileComplete;
```

매개변수	유형	설명
code	int	code 0: 녹음 완료
filepath	string	저장된 녹음 파일의 경로
fileid	string	파일의 url 경로, 90일 동안 서버에 보관합니다

에러 코드

에러 코드 값	원인	솔루션 제안
12289	파일 다운로드 시 파일에 액세스하는 동안 오류가 발생했습니다	파일 경로가 유효한지 확인합니다
12290	서명 확인에 실패했습니다	인증키가 맞는지, 음성 메시지 및 음성 변환 기능이 초기화 되었는지 확인합니다
12291	네트워크 스토리지 시스템 예외입니다	서버가 오디오 파일을 가져오지 못했습니다. API 매개변수 fileid가 올바른지, 네트워크가 정상인지, COS에 파일이 존재하는지 확인합니다
12292	서버 파일 시스템 오류입니다	장치가 인터넷에 액세스할 수 있는지, 파일이 서버에 존재하는지 확인합니다

12293	다운로드 매개변수를 가져오는 과정에서 HTTP 네트워크가 실패했습니다	장치가 인터넷에 액세스할 수 있는지 확인합니다
12294	다운로드 매개변수를 가져오는 과정에서 반환된 패킷이 비어 있습니다	장치가 인터넷에 액세스할 수 있는지 확인합니다
12295	다운로드 매개변수를 가져오는 과정에서 반환된 패킷을 디코딩하지 못했습니다	장치가 인터넷에 액세스할 수 있는지 확인합니다
12297	appinfo가 설정되지 않았습니다	인증키가 맞는지, 음성 메시지 및 음성 변환 기능이 초기화 되었는지 확인합니다

예시 코드

```
//이벤트 수신
ITMGContext.GetInstance().GetPttCtrl().OnDownloadFileComplete +=new QAVDownloadFile
//수신한 이벤트 처리
void OnDownloadFileComplete(int code, string filepath, string fileid){
    //오디오 파일 다운로드 완료 콜백
}
```

오디오 재생

이 API는 오디오를 재생하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT PlayRecordedFile(string filePath)
ITMGPTT PlayRecordedFile(string filePath,int voiceType);
```

매개변수	유형	설명
filePath	string	로컬 오디오 파일 경로
voicetype	int	음성 변조 유형. 자세한 내용은 음성 변조 통합 문서를 참고하십시오.

에러 코드

에러 코드 값	원인	솔루션 제안
20485	재생이 시작되지 않습니다	파일의 존재와 파일 경로의 유효성을 확인합니다

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().PlayRecordedFile(filePath);
```

오디오 재생 콜백

오디오 파일이 재생될 때 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.

API 프로토타입

```
public delegate void QAVPlayFileCompleteCallback(int code, string filepath);
public abstract event QAVPlayFileCompleteCallback OnPlayFileComplete;
```

매개변수	유형	설명
code	int	code 0: 재생 완료
filepath	string	저장된 녹음 파일의 경로

에러 코드

에러 코드 값	원인	솔루션 제안
20481	초기화 오류입니다	장치가 사용 중인지, 권한이 정상인지, 초기화가 정상인지 확인합니다
20482	재생 중에 중단 및 다음 파일 재생을 시도했지만 실패했습니다(정상적으로 성공해야 함)	코드 로직이 올바른지 확인합니다
20483	매개변수가 비어 있습니다	코드의 API 매개변수가 올바른지 확인합니다
20484	내부 오류입니다	플레이어를 초기화하는 동안 오류가 발생했습니다. 이 오류 코드는 일반적으로 디코딩 실패로 인해 발생하며 로그를 통해 문제를 확인해야 합니다.

예시 코드

```
//이벤트 수신:
ITMGContext.GetInstance().GetPttCtrl().OnPlayFileComplete +=new  QAVPlayFileComplet
//수신한 이벤트 처리:
void OnPlayFileComplete(int code, string filepath){
    //오디오 재생 콜백
```

```
}
```

오디오 재생 중지

이 API는 오디오 재생을 중지하는 데 사용됩니다. 재생이 중지되면 재생 완료 콜백이 발생합니다.

API 프로토타입

```
ITMGPTT int StopPlayFile()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().StopPlayFile();
```

오디오 파일 사이즈 획득

이 API는 오디오 파일의 크기를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGPTT GetFileSize(string filePath)
```

매개변수	유형	설명
filePath	String	오디오 파일의 로컬 경로입니다

예시 코드

```
int fileSize = ITMGContext.GetInstance().GetPttCtrl().GetFileSize(filepath);
```

오디오 파일 길이 가져오기

이 API는 오디오 파일의 길이를 밀리초 단위로 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGPTT int GetVoiceFileDuration(string filePath)
```

매개변수	유형	설명
filePath	String	오디오 파일의 로컬 경로입니다

예시 코드

```
int fileDuration = ITMGContext.GetInstance().GetPttCtrl().GetVoiceFileDuration(file
```

빠른 녹음-텍스트 변환

API	API 설명
SpeechToText	음성-텍스트 변환

오디오 파일을 텍스트로 변환

이 API는 지정된 오디오 파일을 텍스트로 변환하는 데 사용됩니다.

API 프로토타입

```
ITMGPTT int SpeechToText(String fileID)
```

매개변수	유형	설명
fileID	String	오디오 파일 url

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().SpeechToText(fileID);
```

오디오 파일을 지정된 언어의 텍스트로 번역

이 API는 인식할 언어를 지정하거나 음성으로 인식된 정보를 지정된 언어로 번역하여 반환할 수 있습니다.

주의사항

번역은 추가 요금이 발생합니다. 자세한 내용은 구매 가이드를 참고하십시오.

API 프로토타입

```
ITMGPTT int SpeechToText(String fileID,String speechLanguage)
ITMGPTT int SpeechToText(String fileID,String speechLanguage,String translatelangua
```

매개변수	유형	설명
fileID	String	90일 동안 서버에 보관되는 오디오 파일의 url
speechLanguage	String	언어 매개변수를 지정하여 해당 언어를 인식합니다. 매개변수는 Language Parameter Reference List를 참고하십시오.

translatelanguage	String	언어 매개변수를 지정하여 해당 언어로 번역합니다. 매개변수는 Language Parameter Reference List를 참고하십시오.

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().SpeechToText(fileID,"cmn-Hans-CN","cmn-Hans-
```

인식 콜백

지정된 오디오 파일이 인식되어 텍스트로 변환되면 메시지를 전달하기 위해 델리게이트 함수를 통해 콜백이 실행됩니다.

API 프로토타입

```
public delegate void QAVSpeechToTextCallback(int code, string fileid, string result
public abstract event QAVSpeechToTextCallback OnSpeechToTextComplete;
```

매개변수	유형	설명
code	int	code 0: 녹음 완료
fileid	string	90일 동안 서버에 보관되는 녹음 파일의 url
result	string	변환된 텍스트

에러 코드

에러 코드 값	원인	솔루션 제안
32769	내부 오류입니다	로그를 분석하고 백엔드에서 클라이언트로 반환된 실제 오류 코드를 얻은 다음 백엔드 담당자에게 도움을 요청합니다
32770	네트워크에 실패했습니다	장치가 인터넷에 액세스할 수 있는지 확인합니다
32772	반환된 패킷을 디코딩하지 못했습니다	로그를 분석하고 백엔드에서 클라이언트로 반환된 실제 오류 코드를 얻은 다음 백엔드 담당자에게 도움을 요청합니다
32774	appinfo가 설정되지 않았습니다	인증키가 맞는지, 음성 메시지 및 음성 변환 기능이 초기화되었는지 확인합니다
32776	authbuffer 확인에 실패했습니다	authbuffer가 올바른지 확인합니다

32784	잘못된 음성-텍스트 변환 매개변수입니다	코드의 API 매개변수 fileid가 비어 있는지 확인합니다
32785	음성-텍스트 변환 및 번역에서 오류가 반환되었습니다	음성 메시지 백엔드 오류입니다. 백엔드에서 클라이언트로 반환된 실제 오류 코드를 받고 백엔드 담당자에게 도움을 요청하십시오.
32787	음성-텍스트 변환에 성공했지만 텍스트 번역 서비스가 활성화되지 않았습니다	콘솔에서 텍스트 번역 서비스를 활성화해야 합니다
32788	음성-텍스트 변환에 성공했지만 텍스트 번역 서비스의 언어 매개변수가 잘못되었습니다	전달된 매개변수를 확인합니다

예시 코드

```
//이벤트 수신
ITMGContext.GetInstance().GetPttCtrl().OnSpeechToTextComplete += new QAVSpeechToTex
//수신한 이벤트 처리
void OnSpeechToTextComplete(int code, string fileid, string result){
    //콜백 인식
}
```

음성 메시지 볼륨 레벨 API

API	API 설명
GetMicLevel	실시간 마이크 볼륨 레벨 가져오기
SetMicVolume	녹음 볼륨 레벨 설정
GetMicVolume	녹음 볼륨 레벨 가져오기
GetSpeakerLevel	실시간 스피커 볼륨 가져오기
SetSpeakerVolume	재생 볼륨 레벨 설정
GetSpeakerVolume	재생 볼륨 레벨 가져오기

음성 메시지의 실시간 마이크 볼륨 가져오기

이 API는 실시간 마이크 볼륨을 가져오는 데 사용됩니다. int 유형 값이 반환됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGPTT int GetMicLevel()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().GetMicLevel();
```

음성 메시지의 녹음 볼륨 설정

이 API는 음성 메시지의 녹음 볼륨을 설정하는 데 사용됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGPTT int SetMicVolume(int vol)
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().SetMicVolume(100);
```

음성 메시지의 녹음 볼륨 가져오기

이 API는 음성 메시지의 녹음 볼륨을 가져오는 데 사용됩니다. int 유형 값이 반환됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGPTT int GetMicVolume()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().GetMicVolume();
```

음성 메시지의 실시간 스피커 볼륨 가져오기

이 API는 실시간 스피커 볼륨을 가져오는 데 사용됩니다. int 유형 값이 반환됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGPTT int GetSpeakerLevel()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().GetSpeakerLevel();
```

음성 메시지 재생 볼륨 설정

이 API는 음성 메시지의 재생 볼륨을 설정하는 데 사용됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGPTT int SetSpeakerVolume(int vol)
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().SetSpeakerVolume(100);
```

음성 메시지의 재생 볼륨 가져오기

이 API는 음성 메시지의 재생 볼륨을 가져오는 데 사용됩니다. int 유형 값이 반환됩니다. 값 범위: 0 - 200.

API 프로토타입

```
ITMGPTT int GetSpeakerVolume()
```

예시 코드

```
ITMGContext.GetInstance().GetPttCtrl().GetSpeakerVolume();
```

고급 API

버전 번호 가져오기

이 API는 분석을 위한 SDK 버전 번호를 가져오는 데 사용됩니다.

API 프로토타입

```
ITMGContext  abstract string GetSDKVersion()
```

예시 코드

```
ITMGContext.GetInstance().GetSDKVersion();
```

로그 출력 레벨 설정

이 API는 출력할 로그의 수준을 설정하는 데 사용되며 초기화 전에 호출해야 합니다. 기본 수준을 유지하는 것이 좋습니다.

API 프로토타입

```
ITMGContext  SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

매개변수 설명

매개변수	유형	설명
levelWrite	ITMG_LOG_LEVEL	기록할 로그 수준을 설정합니다. TMG_LOG_LEVEL_NONE은 쓰지 않음을 나타냅니다. 기본값: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	출력할 로그의 수준을 설정합니다. TMG_LOG_LEVEL_NONE은 출력하지 않음을 나타냅니다. 기본값: TMG_LOG_LEVEL_ERROR

ITMG_LOG_LEVEL은 아래와 같습니다.

ITMG_LOG_LEVEL	설명
TMG_LOG_LEVEL_NONE	로그를 출력하지 않음
TMG_LOG_LEVEL_ERROR	오류 로그 출력(기본값)
TMG_LOG_LEVEL_INFO	정보 로그 출력
TMG_LOG_LEVEL_DEBUG	디버깅 로그 출력
TMG_LOG_LEVEL_VERBOSE	고빈도 로그 출력

예시 코드

```
ITMGContext.GetInstance().SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

로그 출력 경로 설정

이 API는 로그 출력 경로를 설정하는 데 사용됩니다. 기본 경로는 다음과 같습니다. Init 전에 호출해야 합니다.

플랫폼	경로
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents

Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API 프로토타입

```
ITMGContext  SetLogPath(string logDir)
```

매개변수	유형	설명
logDir	String	경로

예시 코드

```
ITMGContext.GetInstance().SetLogPath(path);
```

프로젝트 내보내기

최종 업데이트 날짜: : 2024-01-18 16:05:38

본문은 Unity용 Tencent Cloud Game Multimedia Enginec(GME) API에 대한 Unity 프로젝트 내보내기를 구성하는 방법을 설명합니다.

iOS용으로 내보내기

Unity 프로젝트에서 Xcode 프로젝트로 내보낼 때 GME 동적 라이브러리를 처리해야 하며 처리 방법은 Unity 버전에 따라 다릅니다.

1. 동적 라이브러리 처리(Unity 2019 이상 버전)

구성 원리

Editor OnPostprocessBuild 스크립트를 생성하고 UnityEditor.iOS.Xcode.Extensions.PBXProjectExtensions.AddFileToEmbedFrameworks를 사용하면 이 API가 자동으로 동적 라이브러리를 최종 패키지 번들의 프레임워크 디렉터리에 복사하고 서명합니다.
비즈니스 레이어는 필요한 기능에 따라 동적 라이브러리를 삭제할 수 있으며 동적 라이브러리 목록에 따라 예시 코드에서 가져온 프레임워크 목록을 결정할 수 있습니다. 동적 라이브러리 기능은 동적 라이브러리 디렉터리를 참고하십시오.

```
string[] framework_names = {
    "libgme_fdkaac.framework",
    "libgme_lamemp3.framework",
    "libgme_ogg.framework",
    "libgme_soundtouch.framework"
};
```

예시 코드

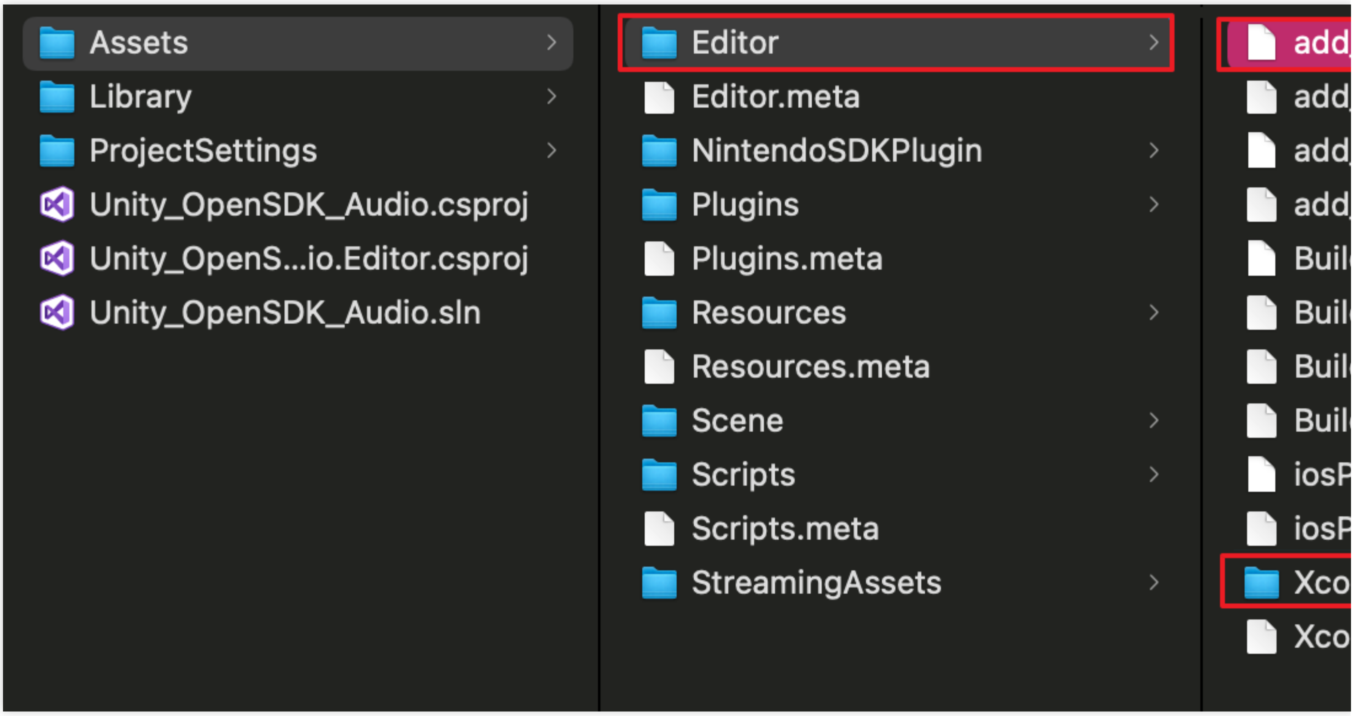
Demo 프로젝트의 add_dylib.cs 스크립트 파일을 참고하여 자신의 프로젝트 요구 사항에 따라 이 부분의 코드를 프로젝트의 Editor 폴더 아래에 넣을 수 있습니다.

```
[UnityEditor.Callbacks.PostProcessBuild(1002)]
public  static void OnPostprocessBuild (UnityEditor.BuildTarget BuildTarget, string
    if (BuildTarget == UnityEditor.BuildTarget.iOS) {
        UnityEngine.Debug.Log ("OnPostprocessBuild add_dylib:" + path);
#if UNITY_EDITOR_OSX || UNITY_STANDALONE_OSX
        {
            string projPath = UnityEditor.iOS.Xcode.PBXProject.GetPBXProjectPath (p
            UnityEditor.iOS.Xcode.PBXProject proj = new UnityEditor.iOS.Xcode.PBXPr
```

```
            proj.ReadFromString (System.IO.File.ReadAllText (projPath));
            // string targetGuid = proj.TargetGuidByName (UnityEditor.iOS.Xcode.PBX
            string targetGuid = proj.GetUnityMainTargetGuid();  // 2019

            // 가져오기한 framework에 따라 삭제
            string[] framework_names = {
                "libgme_fdkaac.framework",
                "libgme_lamemp3.framework",
                "libgme_ogg.framework",
                "libgme_soundtouch.framework"
            };

            for (int i = 0; i < framework_names.Length; i++)
            {
                string framework_name = framework_names[i];
                string dylibGuid = null;
                dylibGuid = proj.FindFileGuidByProjectPath("Frameworks/Plugins/iOS/

                if (dylibGuid == null) {
                    UnityEngine.Debug.LogWarning (framework_name + " guid not found
                } else {
                    UnityEngine.Debug.LogWarning (framework_name + " guid:" + dylib
                    // proj.AddDynamicFramework (targetGuid, dylibGuid);
                    UnityEditor.iOS.Xcode.Extensions.PBXProjectExtensions.AddFileTo

                    proj.AddBuildProperty(targetGuid, "LD_RUNPATH_SEARCH_PATHS", "@
                    System.IO.File.WriteAllText (projPath, proj.WriteToString ());
                }
            }
        }
#endif
    }
}
```

2. 동적 라이브러리 처리(Unity 2019 이하 버전)

현재 Unity 2019 이상 버전에서만 UnityEditor.iOS.Xcode.Extensions를 사용할 수 있으며, Unity 이전 버전인 경우 Unity 상위 버전에서 하위 버전 Unity로 UnityEditor.iOS.Xcode 패키지를 내보내거나 UnityEditorAV.iOS.XCode.zip 첨부 파일을 직접 참고하여 이 파일의 압축을 풀어 프로젝트 디렉터리의 Editor 폴더 아래에 넣습니다.

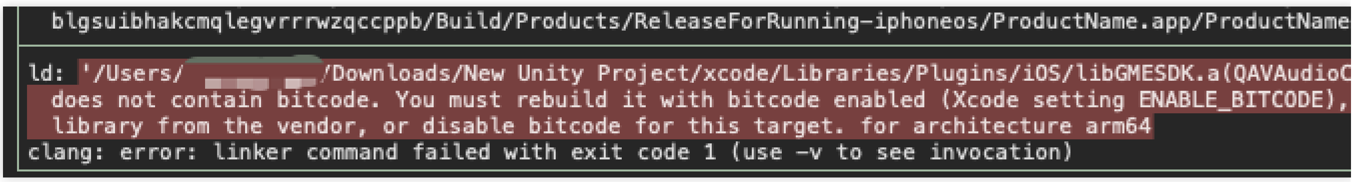


3. Xcode 프로젝트 내보내기

Xcode 버전이 10.0 이상인지 확인하고 Unity 에디터에서 Xcode 프로젝트를 내보냅니다.

4. BitCode 비활성화

컴파일 중에 다음 오류가 발생하면 Bitcode를 비활성화하십시오. **Targets**>**Build Settings**에서 Bitcode를 검색하고 해당 옵션을 NO로 설정합니다.



5. iOS에 대한 액세스 추가

Required background modes: 백그라운드에서 실행을 허용합니다(선택 사항).
Microphone Usage Description: 마이크에 대한 액세스를 허용합니다.

6. 라이브러리 파일 추가

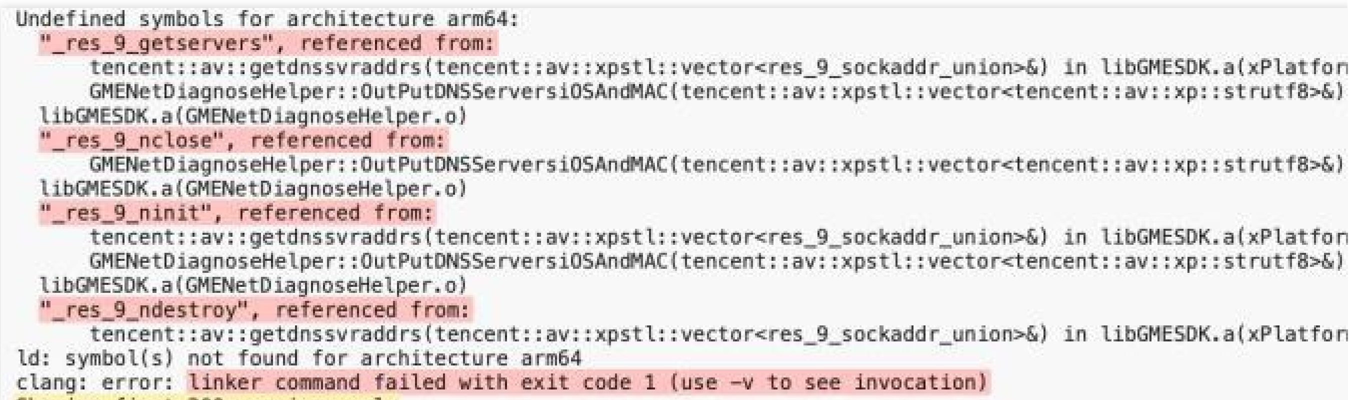
컴파일 중 아래와 같은 오류가 발생하면 라이브러리 파일을 완성하십시오.

라이브러리 파일 목록은 다음과 같습니다.

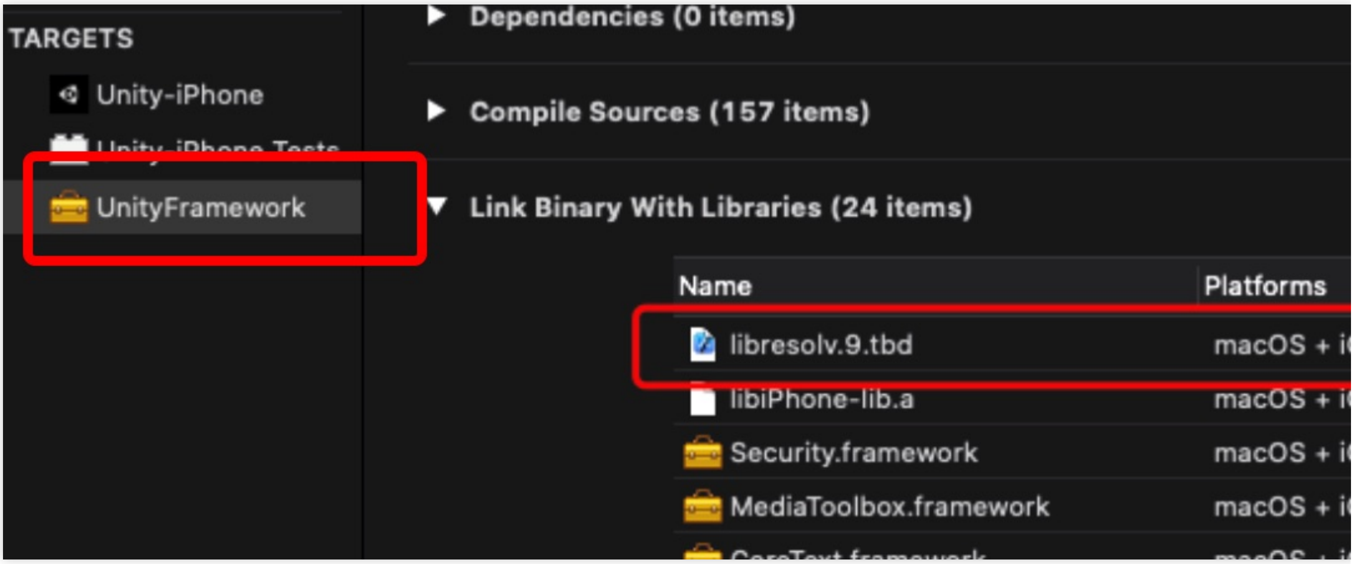
```
libc++.tbd
CoreMedia.framework
libresolv.tbd
AVFoundation.framework
Security.framework
CoreAudio.framework
AudioToolbox.framework
libiconv.tbd
libz.tbd
SystemConfiguration.framework
OpenAL.framework
```

7. libresolv9.tbd 추가

다음 오류가 발생하는 경우:



UnityFramework에 libresolv9.tbd를 추가합니다.



8. 내보내기 관련 FAQ

내보내기 관련 문제는 iOS용으로 내보내기를 참고하십시오.

Android용으로 내보내기

1. 불필요한 lib 파일 삭제

Unity용 GME SDK는 기본적으로 arm64-v8a, armeabi-v7a 및 x86용 lib 파일을 제공합니다. 필요에 따라 불필요한 파일을 삭제하십시오.

아키텍처 손실

내보낸 Android 실행 파일에 지정된 아키텍처가 없으면 Crash가 발생합니다.

실행 가능한 apk 파일을 내보낸 후 열 때 검은 화면이나 충돌이 발생하는 원인은 일반적으로 해당 아키텍처 lib 파일이 없기 때문입니다. 프로젝트에 따라 해당 아키텍처 lib 파일을 추가하거나 삭제하십시오.

2. 권한 구성

2.1 필수 권한

프로젝트의 AndroidManifest.xml 파일에 다음 권한을 추가합니다:

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

2.2 필요에 따라 권한 추가

필요에 따라 프로젝트의 AndroidManifest.xml 파일에 다음 권한을 추가합니다:

읽기/쓰기 권한

블루투스 권한

읽기/쓰기 권한이 필요하지 않습니다. 다음 규칙에 따라 추가 여부를 결정합니다.

기본 로그 경로(/sdcard/Android/data/xxx.xxx.xxx/files)를 사용하는 경우 SetLogPath를 호출하지 않으며 WRITE_EXTERNAL_STORAGE 권한이 필요하지 않음을 의미합니다.

setLogPath API를 호출하여 로그 경로를 외부 저장 장치로 설정하고 음성 메시지 녹음의 저장 경로가 외부 저장 장치인 경우 사용자에게 WRITE_EXTERNAL_STORAGE 권한을 신청하고 사용자의 승인을 받아야 합니다.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

다음 규칙에 따라 블루투스 권한을 추가하십시오.

프로젝트의 targetSDKVersion이 v30 이하인 경우:

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

프로젝트의 targetSDKVersion이 v31 이상인 경우:

```
<uses-permission android:name="android.permission.BLUETOOTH" android:maxSdkVersion=
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

3. 내보내기 관련 FAQ

내보내기 관련 문제는 Android용으로 내보내기를 참고하십시오.

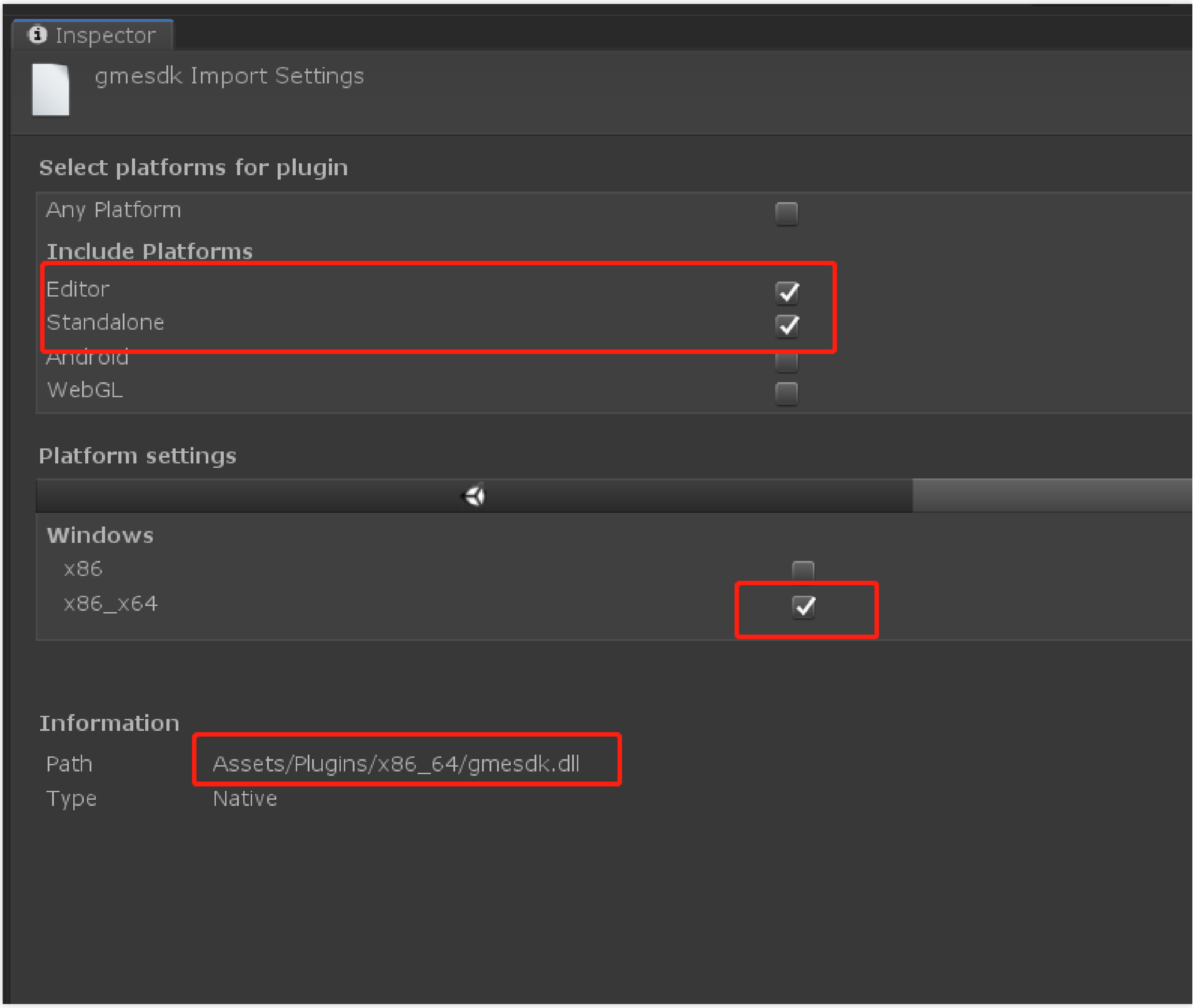
Windows용으로 내보내기

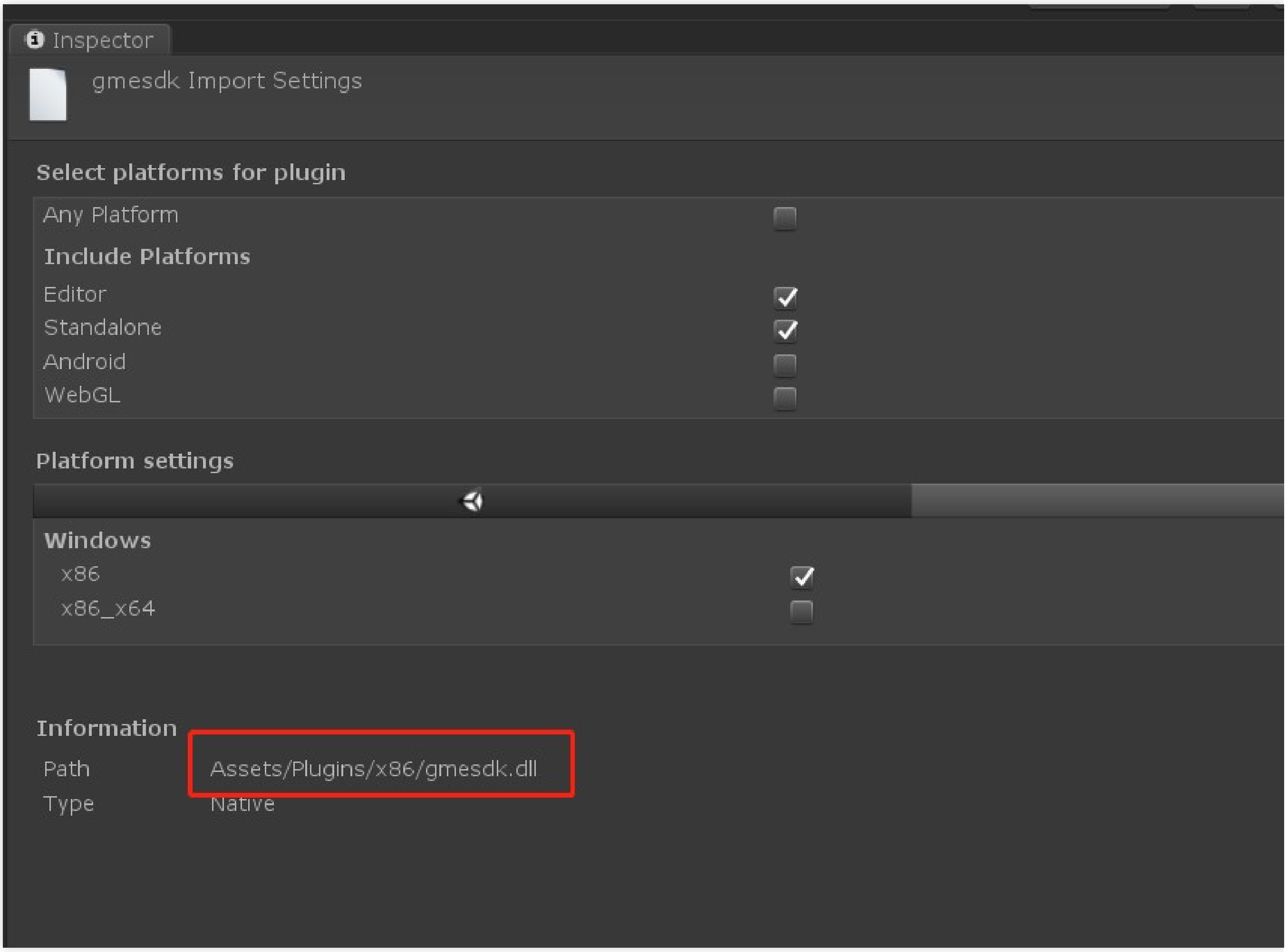
내보내기 관련 문제는 Windows용으로 내보내기를 참고하십시오.

WebGL용으로 내보내기

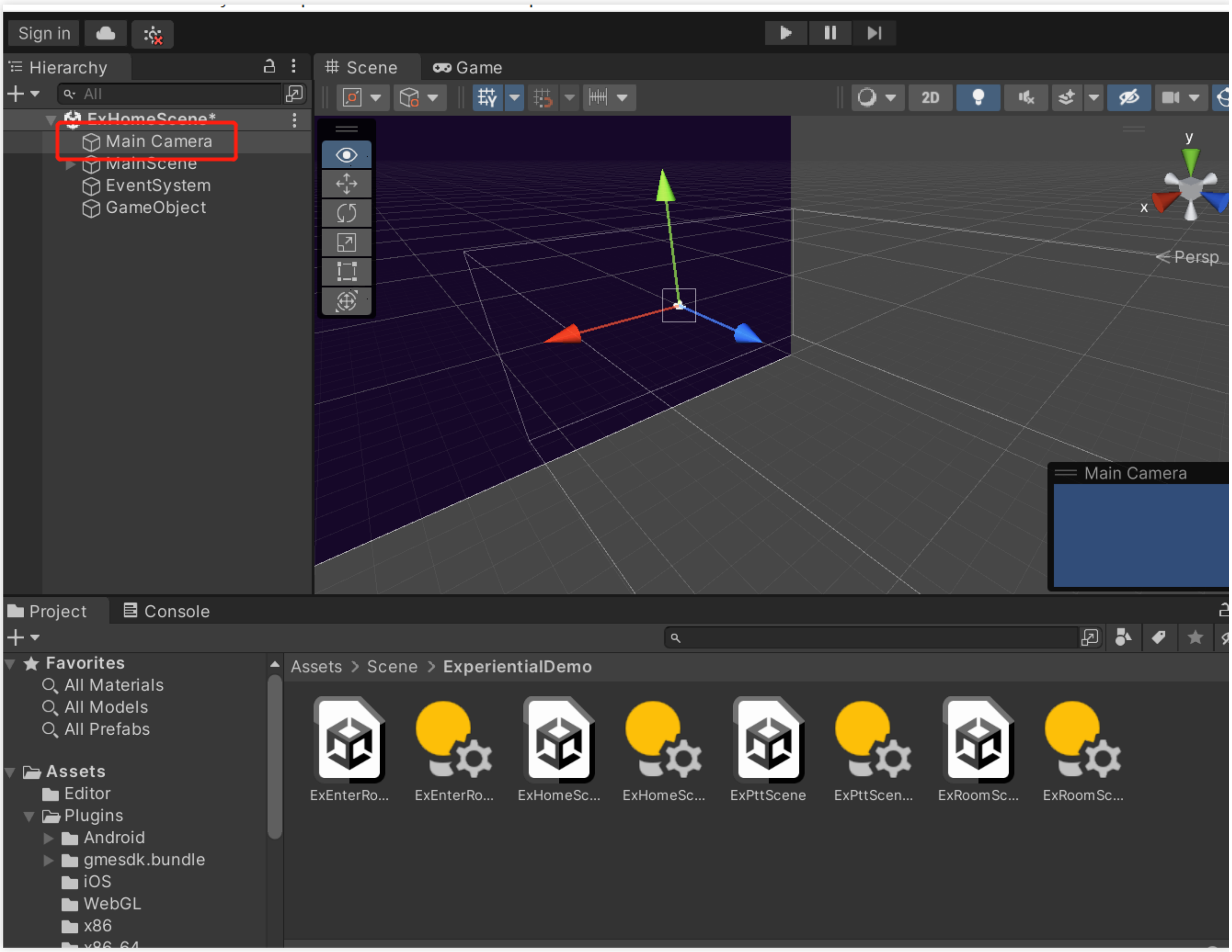
1. WebGL에서 plugins 구성

WebGL 플랫폼에서 gmesdk와의 충돌을 피하기 위해 Windows 플랫폼에서 gmesdk.dll의 적용 범위를 설정합니다.


Inspector
gmesdk Import Settings
Select platforms for plugin
Any Platform
Include Platforms
Editor
Standalone
Android
WebGL
Platform settings
Windows
x86
x86_x64
Information
Path
Assets/Plugins/x86_64/gmesdk.dll
Type
Native



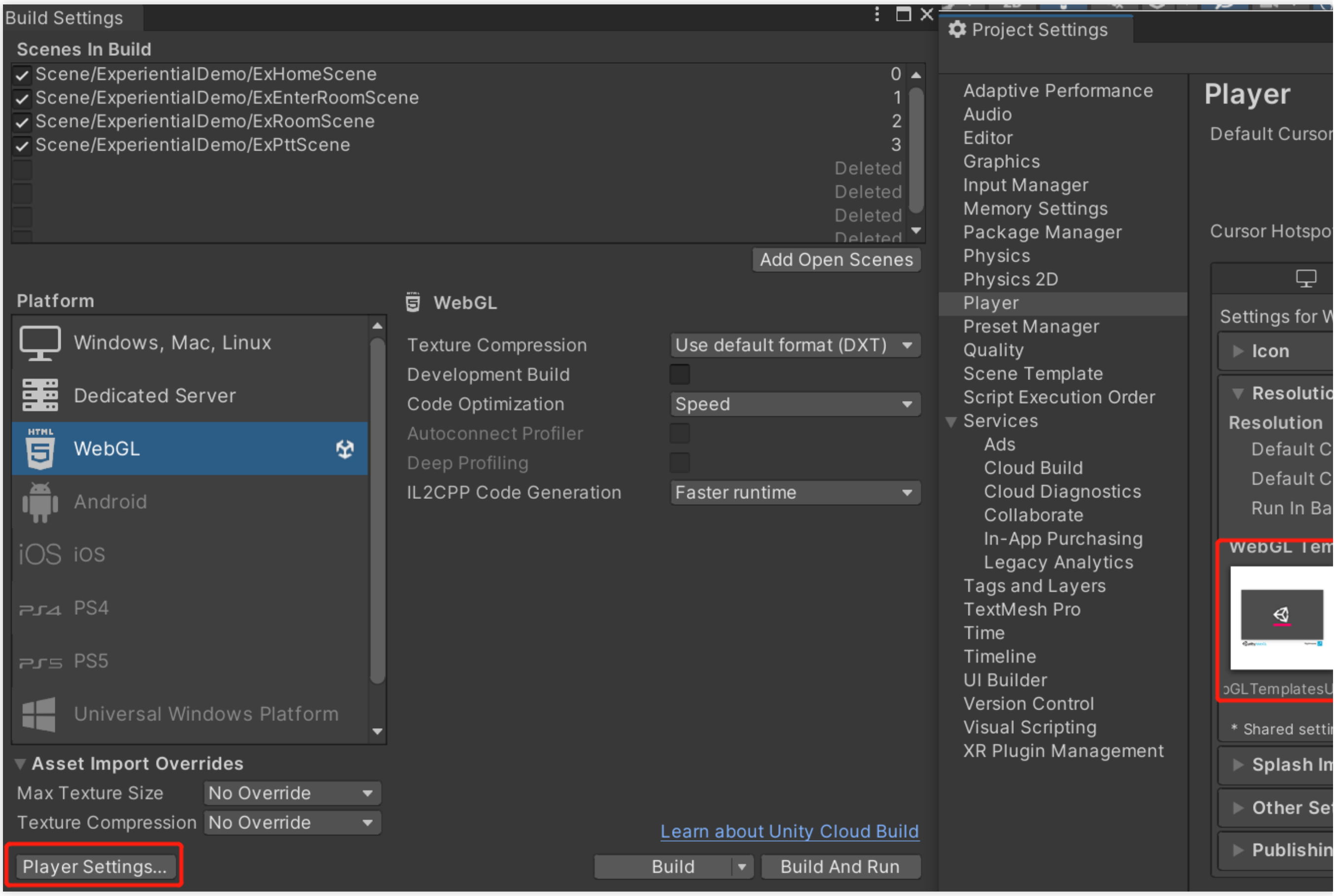
2. Flare Layer 취소(Unity 2018 이상 버전)



일부 Unity 버전은 MainCamera의 Flare Layer 모드를 더 이상 지원하지 않으므로 패키징할 Scene에서 Flare Layer를 선택 취소해야 합니다. 그렇지 않으면 다음 오류가 보고됩니다.

3. 템플릿 선택

WebGL용으로 내보내기 할 때 GME WebGL 템플릿을 선택해야 패키징된 아티팩트가 관련 종속 라이브러리를 올바르게 가져옵니다. 기본적으로 프로젝트는 Unity2018 및 Unity2019 버전을 지원하는 GMEWebGLTemplatesUnity2018 템플릿을 사용합니다. Unity2020 및 Unity2021 버전의 경우 패키징할 때 사용하는 템플릿을 변경하고 패키징에 GMEWebGLTemplatesUnity2021을 사용해야 합니다.



4. 프런트엔드 라이브러리 가져오기

GME-WebGL을 자신의 프로젝트로 가져오고 Unity를 사용하여 해당 웹 페이지를 생성할 때 프런트 엔드 라이브러리를 수동으로 가져와 프런트 엔드 라이브러리 파일을 해당 참조 위치에 배치하고 오디오 태그를 추가해야 합니다(아래 이미지 참고). Unity 아티팩트를 패키징할 때마다 상기 작업을 자동으로 완료하려면 GME-WebGL demo를 참고하여 프로젝트에 해당하는 템플릿을 추가하면 됩니다.

```html
<!DOCTYPE html>
<html lang="en-us">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Unity WebGL Player | %UNITY_WEB_NAME%</title>
    <link rel="shortcut icon" href="TemplateData/favicon.ico">
    <link rel="stylesheet" href="TemplateData/style.css">
    <script src="https://code.jquery.com/jquery-3.5.0.min.js"></script>
    <script src="TemplateData/UnityProgress.js"></script>
    <script src="%UNITY_WEBGL_LOADER_URL%"></script>
    <script src="WebRTCService.js"></script>
    <script src="implementation.js"></script>
    <script>
      var unityInstance = UnityLoader.instantiate("unityContainer", "%UNITY_WEBGL_BUILD_URL%",
    </script>
  </head>
  <body>
    <div id="gme-audio-wrap"></div>
    <div class="webgl-content">
      <div id="unityContainer" style="width: %UNITY_WIDTH%px; height: %UNITY_HEIGHT%px"></div>
      <div class="footer">
        <div class="webgl-logo"></div>
        <div class="fullscreen" onclick="unityInstance.SetFullscreen(1)"></div>
        <div class="title">%UNITY_WEB_NAME%</div>
      </div>
    </div>
  </body>
</html>
```

5. 내보내기 관련 FAQ

내보내기 관련 문제는 Unity-WebGL용으로 내보내기를 참고하십시오.

HD 음질 사용

최종 업데이트 날짜: : 2024-01-18 16:05:38

본문은 Unity 개발자를 위해 Tencent Cloud Game Multimedia Engine(GME) API용 Unity 실시간 음성 채팅 HD 음질 방의 사용 방법을 안내합니다.

전제 조건

GME SDK v2.9부터 GME Unity SDK는 기본적으로 실시간 음성 채팅 HD 음질 방의 사용을 지원하지 않습니다. 버전 2.9 이상이 아닌 경우 다음 작업이 필요하지 않습니다.

관련된 기능

조정하지 않으면 Unity SDK에서 다음 기능이 누락됩니다.
실시간 음성 채팅 HD 음질 방은 음질을 참고하십시오.
반주 기능 재생은 Accompaniment in Voice Chat을 참고하십시오.
실시간 음성 변조 및 음성 메시지 음성 변조 기능은 실시간 음성 효과를 참고하십시오.

SDK 업데이트

다운로드 링크

다운로드 가이드를 통해 Unity SDK의 표준 버전을 다운로드한 경우 다른 라이브러리 파일을 다운로드해야 하므로 티켓을 제출하거나 문의하기을 통해 Tencent Cloud 직원에게 문의하십시오.

라이브러리 파일 교체

기존 GME 라이브러리 파일을 삭제하고 다운로드한 라이브러리 파일을 사용하십시오. 추가된 라이브러리 파일의 예시는 아래 이미지와 같습니다.

라이브러리 파일의 해당 기능

자신의 필요에 따라 해당 라이브러리 파일만 가져올 수 있습니다. 예를 들어 음성 변조 기능만 필요한 경우 libgme_soundtouch만 가져오면 됩니다.

라이브러리 파일	해당 기능
libgme_fdkaac	1. 표준, HD 음질 방 입장 시 사용 2. acc 형식의 반주 파일 재생 시 사용

libgme_faad2	mp4 형식의 반주 파일 재생 시 사용
libgme_ogg	ogg 형식의 반주 파일 재생 시 사용
libgme_lamemp3	mp3 형식의 반주 파일 재생 시 사용
libgme_soundtouch	음성 및 피치 변경에 사용

내보내기 구성

필요한 라이브러리 파일을 구성한 후 내보내기 시 iOS 플랫폼 패키지 동적 라이브러리를 구성해야 하며 다른 플랫폼은 기본적으로 내보내기 할 수 있습니다.

Unity 2019 이상

구성 원리

Editor OnPostprocessBuild 스크립트를 생성하고 UnityEditor.iOS.Xcode.Extensions.PBXProjectExtensions.AddFileToEmbedFrameworks를 사용하면 이 API가 자동으로 동적 라이브러리를 최종 패키지 번들의 프레임워크 디렉터리에 복사하고 서명합니다.

예시 코드

Demo 프로젝트의 add_dylib.cs 스크립트 파일을 참고하여 자신의 프로젝트 요구 사항에 따라 이 부분의 코드를 프로젝트의 Editor 폴더 아래에 넣을 수 있습니다.

```
    [UnityEditor.Callbacks.PostProcessBuild(1002)]
    public  static void OnPostprocessBuild (UnityEditor.BuildTarget BuildTarget, st
        if (BuildTarget == UnityEditor.BuildTarget.iOS) {
            UnityEngine.Debug.Log ("OnPostprocessBuild add_dylib:" + path);
#if UNITY_EDITOR_OSX || UNITY_STANDALONE_OSX
            {
                string projPath = UnityEditor.iOS.Xcode.PBXProject.GetPBXProjectPat
                UnityEditor.iOS.Xcode.PBXProject proj = new UnityEditor.iOS.Xcode.P

                proj.ReadFromString (System.IO.File.ReadAllText (projPath));
                // string targetGuid = proj.TargetGuidByName (UnityEditor.iOS.Xcode
                string targetGuid = proj.GetUnityMainTargetGuid();      // 2019

                // 가져오기한 framework에 따라 삭제
                string[] framework_names = {
                    "libgme_fdkaac.framework",
                    "libgme_lamemp3.framework",
                    "libgme_ogg.framework",
                    "libgme_soundtouch.framework"
```

```
                };

                for (int i = 0; i < framework_names.Length; i++)
                {
                    string framework_name = framework_names[i];
                    string dylibGuid = null;
                    dylibGuid = proj.FindFileGuidByProjectPath("Frameworks/Plugins/

                    if (dylibGuid == null) {
                        UnityEngine.Debug.LogWarning (framework_name + " guid not f
                    } else  {
                        UnityEngine.Debug.LogWarning (framework_name + " guid:" + d
                        // proj.AddDynamicFramework (targetGuid, dylibGuid);
                        UnityEditor.iOS.Xcode.Extensions.PBXProjectExtensions.AddFi

                        proj.AddBuildProperty(targetGuid, "LD_RUNPATH_SEARCH_PATHS"
                        System.IO.File.WriteAllText (projPath, proj.WriteToString (
                    }
                }
            }
#endif
        }
    }
```

Unity 2019보다 낮은 버전

현재 Unity 2019 이상 버전에서만 UnityEditor.iOS.Xcode.Extensions를 사용할 수 있으며, Unity 이전 버전인 경우 Unity 상위 버전에서 하위 버전 Unity로 UnityEditor.iOS.Xcode 패키지를 내보내거나 UnityEditorAV.iOS.XCode.zip 첨부 파일을 직접 참고하여 이 파일의 압축을 풀어 프로젝트 디렉터리의 Editor 폴더 아래에 넣습니다.

SDK for Unreal Engine
Integrating SDK

최종 업데이트 날짜: : 2024-01-18 15:02:24

Overview

This document describes how to configure a Unreal Engine project for the GME APIs for Unreal Engine.

SDK Download

1. Download the applicable demo and SDK. For more information, see SDK Download Guide.
2. Locate the SDK resources for Unreal Engine on the page.
3. Click **Download**. After decompression, the downloaded SDK resources include the following files:

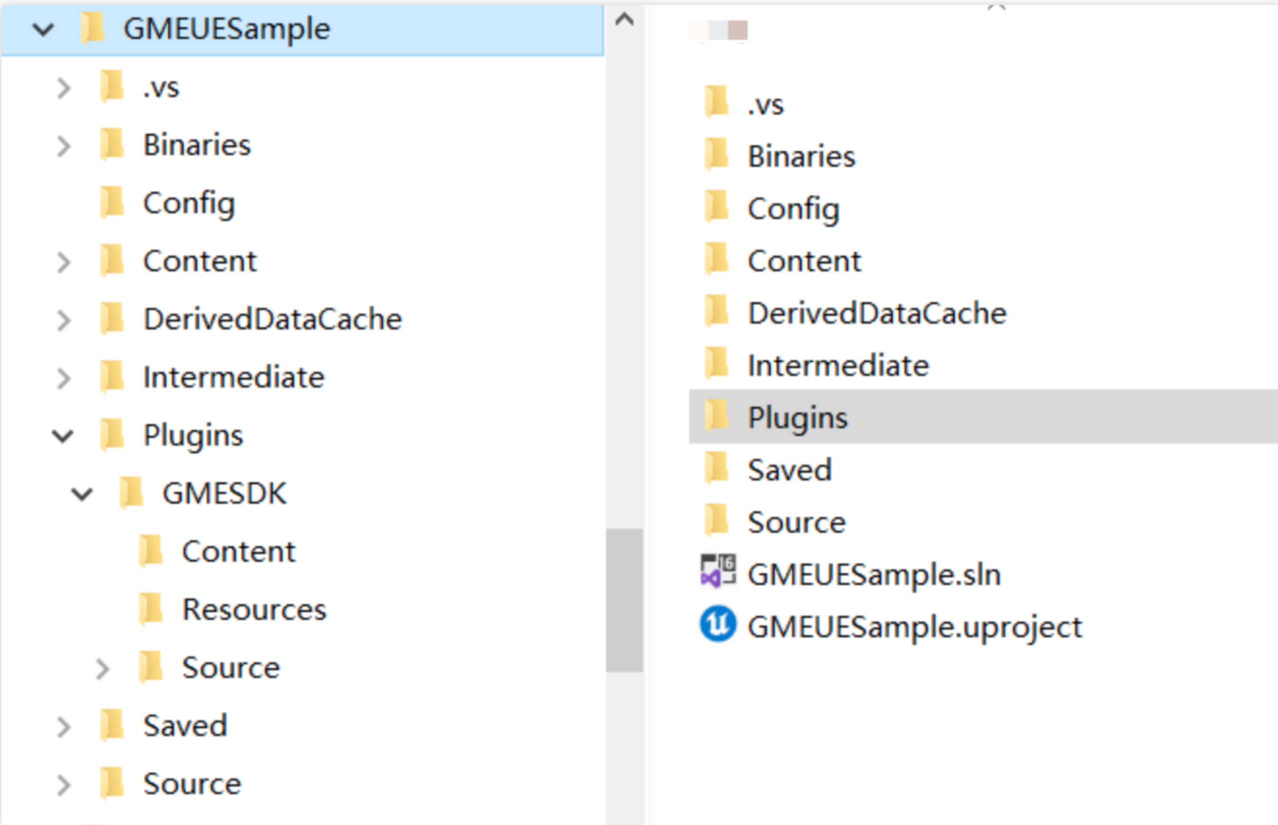
File name	Description	Purpose
GMESDK.uplugin	`.uplugin` file	Plugin configuration file
Resources	Plugin resource file	Plugin resource file
Source	SDK file	SDK library files and code files for various platforms, such as header files

Platforms supported by the SDK for Unreal Engine:
SDK for Unreal Engine has integrated Windows, macOS, Android, and iOS platform architectures. If you need console platform architectures, contact us.

Project Configuration

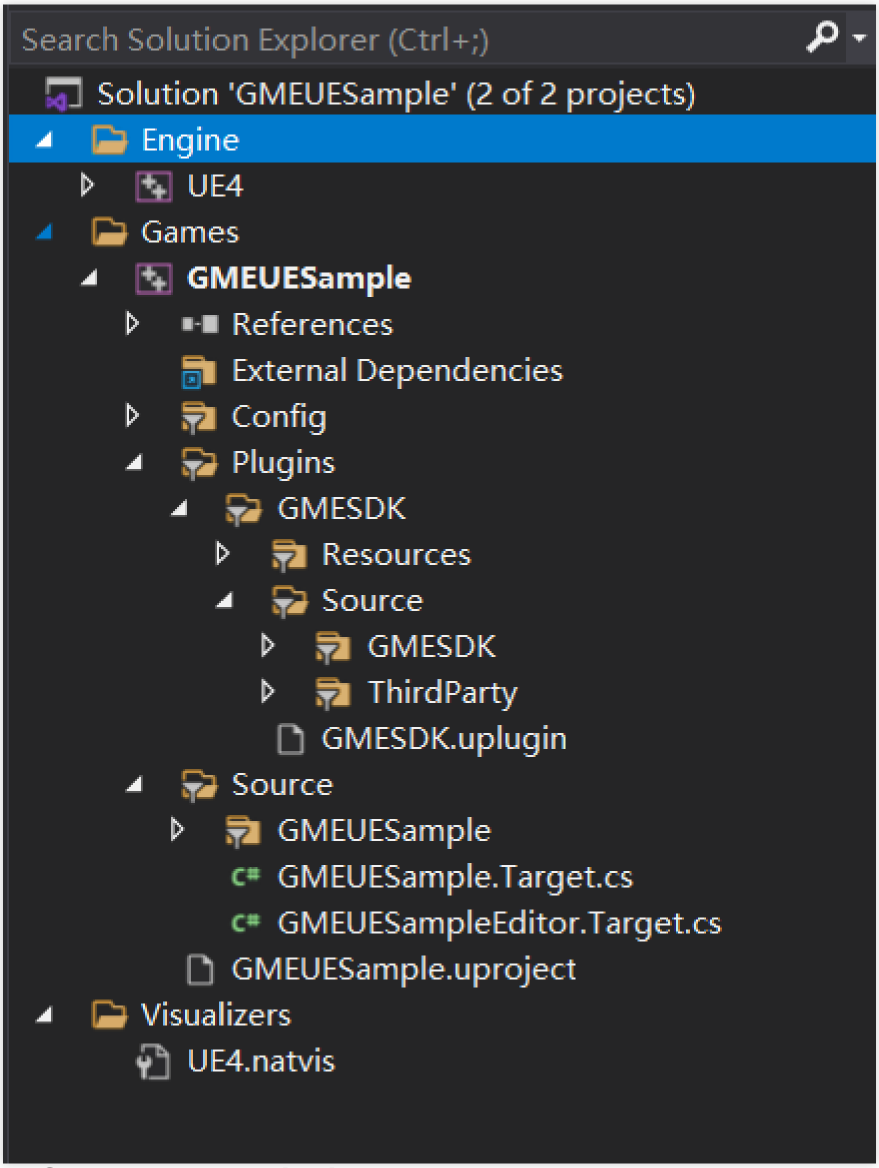
Step 1: Import Plugins files

If no `Plugins` folder exists in the root directory of the game project (the directory of the `*.uproject` file), create one first and copy the GME SDK to the `Plugins` folder. Then, you will see the directory structure as shown below:



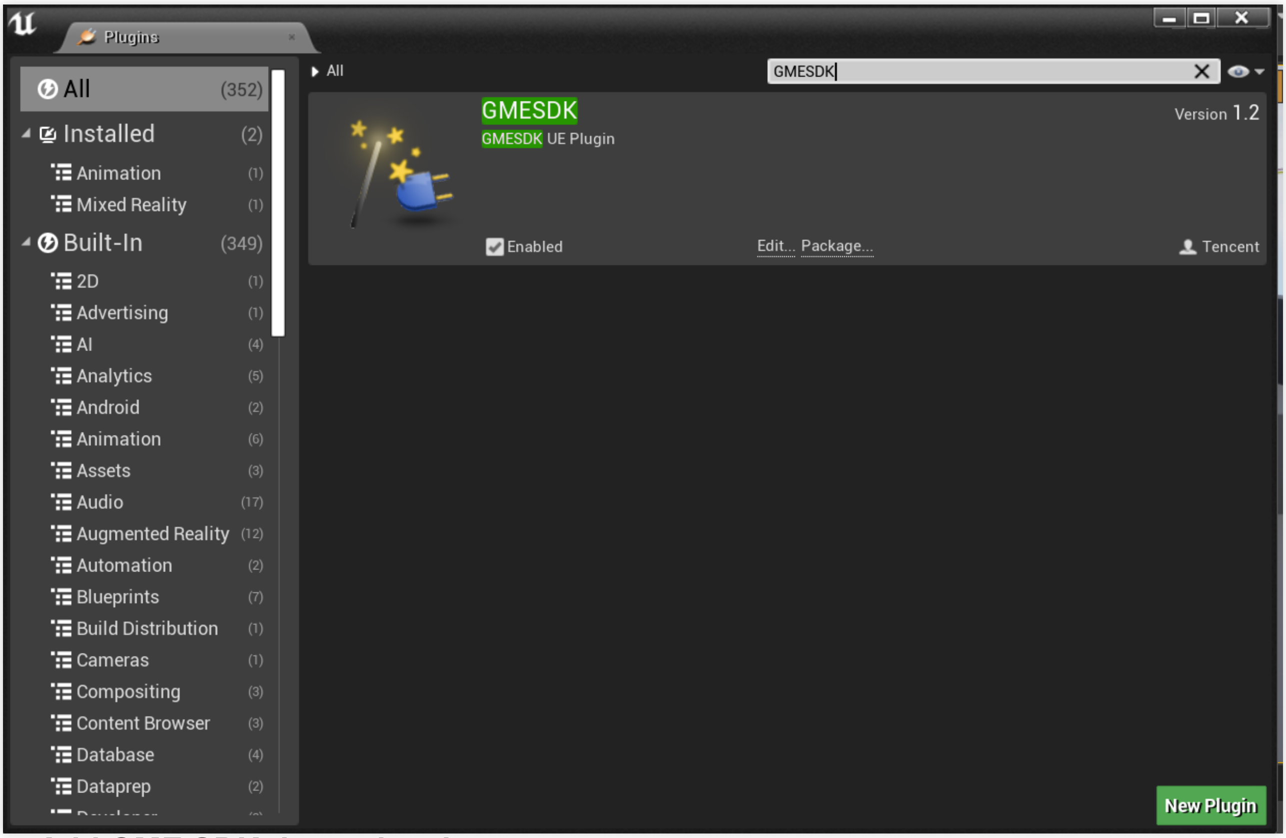
Step 2: Compile the plugin

Refresh the C++ project in Visual Studio/Xcode, open it, and you will see the directory structure like the following. Then, compile the game.



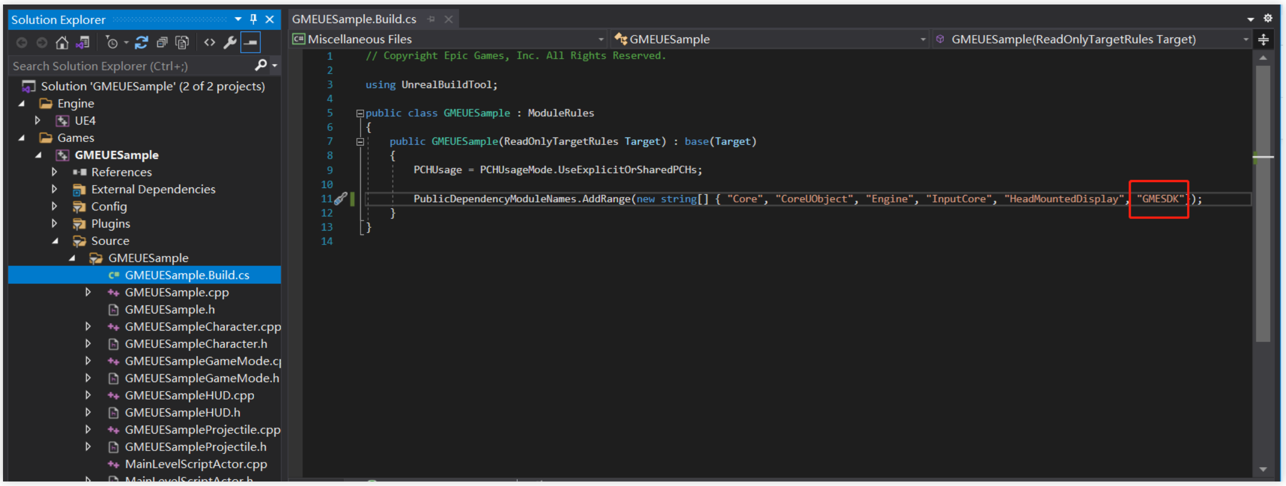
Step 3. Complete compilation

After completing the compilation, restart Unreal Engine Editor. In Unreal Engine Editor, open the plugin manager, and you can see that the GME SDK has been imported into the project. Make sure that the GME SDK is enabled.



Step 4. Add GME SDK dependencies

Add GME SDK dependencies to the `.build.cs` file of the game project.



Adaptations of Different Unreal Engine Versions

Unreal Engine 4.21 and later

If you are using Unreal Engine 4.21 or later, you need to add the following code after downloading the GME sample code for Unreal Engine:

```
AUEDemoLevelScriptActor::AUEDemoLevelScriptActor()
```



```
{
    PrimaryActorTick.bCanEverTick = true;
}
```

Note:

Tick is disabled by default and must be enabled manually.

Unreal Engine 4.26

If you are using Unreal Engine 4.26, you need to download the adaptation file and import it into the project. The downloaded file contains two folders: `Source` and `Plugins` .

For a demo project, import both folders into the project in an overwriting manner.

If you only need the GME SDK, import the `Plugins` folder only.

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:02:24

This document describes how to integrate with and debug GME client APIs for the voice messaging and speech-to-text services for Unreal Engine.

Key Considerations for Using GME

GME provides the real-time voice and voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

Notes

You have created a GME application and obtained the SDK AppID and key. For more information, see Activating Services.

You have activated **GME real-time voice and voice messaging and speech-to-text services**. For more information, see Activating Services.

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error code, see Error Codes.

Note:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.

Non-streaming speech-to-text API ***SpeechToText()***: There can be up to 10 concurrent requests per account.

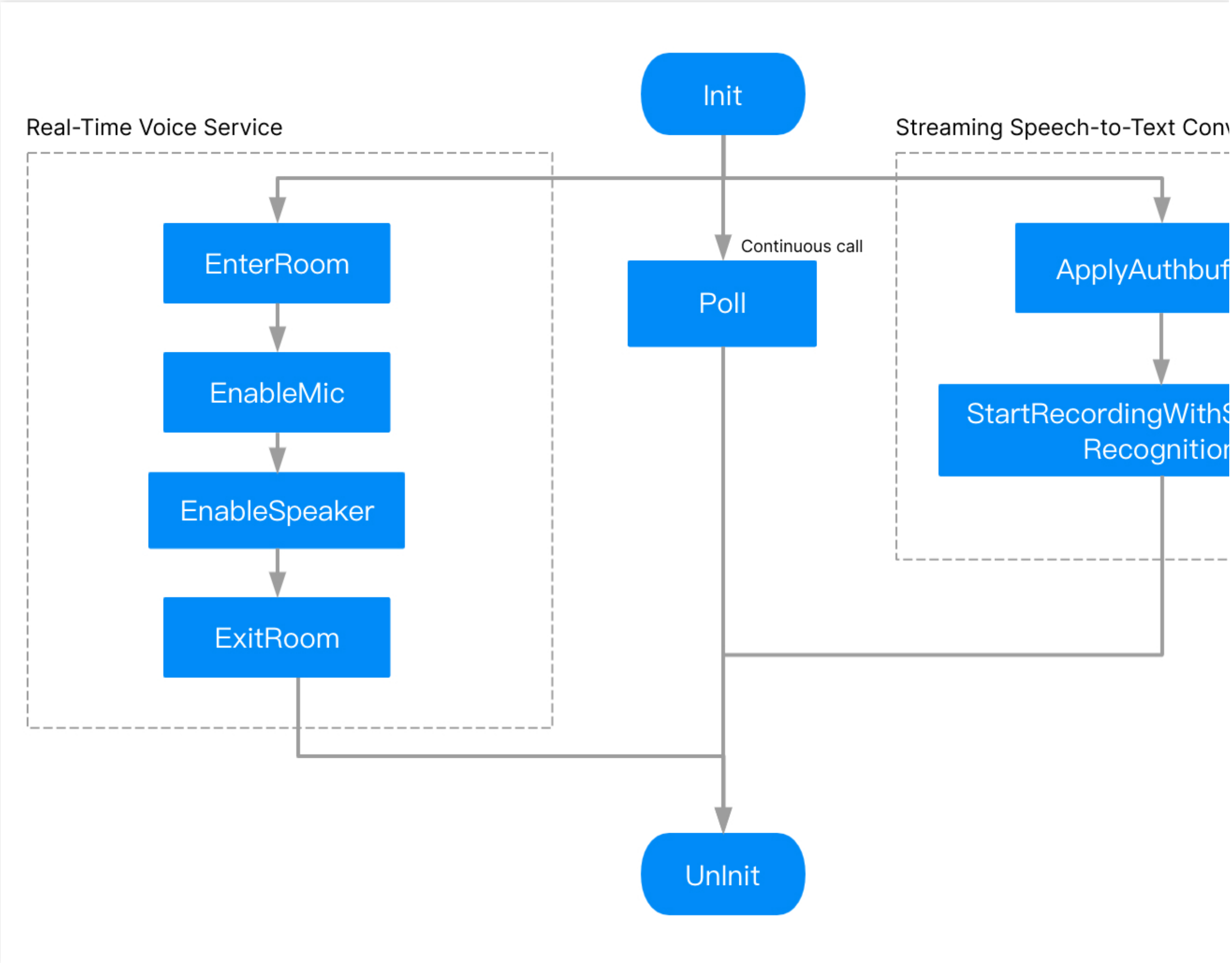
Streaming speech-to-text API ***StartRecordingWithStreamingRecognition()***: There can be up to 50 concurrent requests per account.

Real-time streaming speech-to-text API ***StartRealTimeASR()***: There can be up to 50 concurrent requests per account.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Initializing authentication, API: ApplyPTTAuthbuffer
4. Starting streaming speech recognition, API: StartRecordingWithStreamingRecognition
5. Stop recording, API: StopRecording
6. Uninitializing GME, API: UnInit

C++ classes

Class	Description
ITMGContext	Key APIs
ITMGPTT	Voice messaging and speech-to-text APIs

Key APIs

API	Description
Init	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME

Preparations

You need to import the header file `tmg_sdk.h` first before you can access GME. The classes in the header file inherit `ITMGDelegate` for message delivery and callback.

Sample code

```
#include "tmg_sdk.h"

class UEDEMO1_API AUEDemoLevelScriptActor : public ALevelScriptActor, public ITMGDe
{
public:
...
private:
...
}
```

Setting singleton

You need to get `ITMGContext` first before you can call the `EnterRoom` function, because all calls begin with `ITMGContext` and callbacks are passed to the application through `ITMGDelegate` .

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->SetTMGDelegate(this);
```

Message delivery

The API class uses the `Delegate` method to send callback notifications to the application. `ITMG_MAIN_EVENT_TYPE` indicates the message type. The data on Windows is in json string format. For the key-value pairs, please see the relevant documentation.

Sample code

```
// Function implementation:
//UEDemoLevelScriptActor.h:
class UEDEMO1_API AUEDemoLevelScriptActor : public ALevelScriptActor, public SetTMG
{
public:
    void OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data);
}

//UEDemoLevelScriptActor.cpp:
void AUEDemoLevelScriptActor::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* d
    // Identify and manipulate `eventType` here
}
```

Initializing SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

Parameter	Type	Description
sdkAppId	const char*	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	const char*	`openID` can only be in `Int64` type, which is passed in after being converted to a `const char*`. You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
AV_OK = 0	Initialized SDK successfully.

AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

Notes on 7015 error code

The 7015 error code is judged by md5. If this error is reported during integration, please check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party reinforcement, Unity packaging mechanism and other factors, the md5 of the library file will be affected, resulting in misjudgment. **Please ignore this error in the logic for official release**, and try to avoid displaying it in the UI.

Sample code

```
std::string appid = TCHAR_TO_UTF8(CurrentWidget->editAppID->GetText().ToString().op
std::string userId = TCHAR_TO_UTF8(CurrentWidget->editUserID->GetText().ToString().
ITMGContextGetInstance()->Init(appid.c_str(), userId.c_str());
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. The `Poll` API is GME's message pump and should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run abnormally. For more information, see the `EnginePollHelper` file in SDK Download Guide.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```

Sample code

```
// Declaration in the header file
virtual void Tick(float DeltaSeconds);

void AUEDemoLevelScriptActor::Tick(float DeltaSeconds) {
```

```
    Super::Tick(DeltaSeconds);
    ITMGContextGetInstance()->Poll();
}
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
ITMGContext int Pause()
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
ITMGContext int Resume()
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

API prototype

```
ITMGContext int Uninit()
```

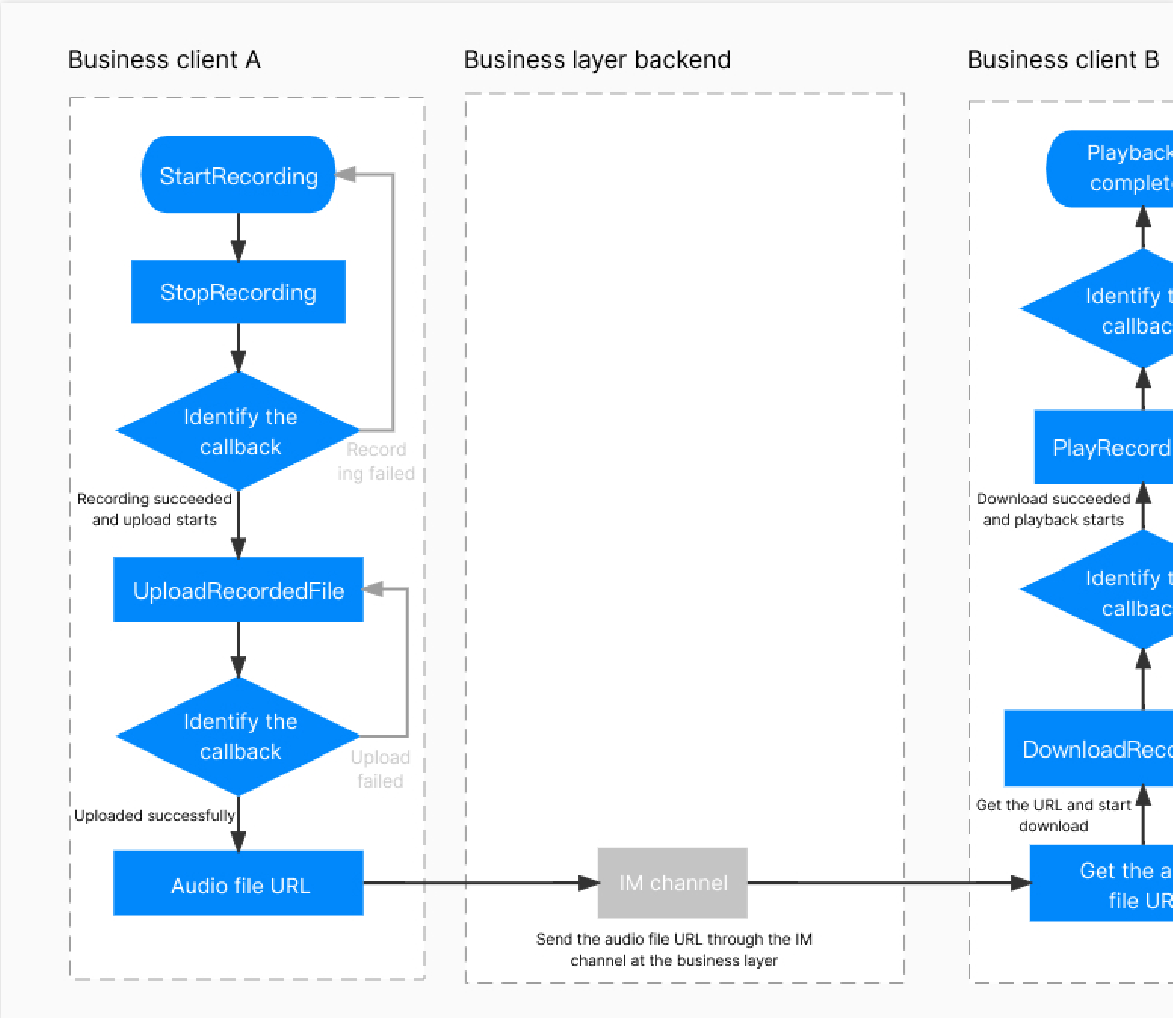
Voice Messaging and Speech-to-Text Services

Note:

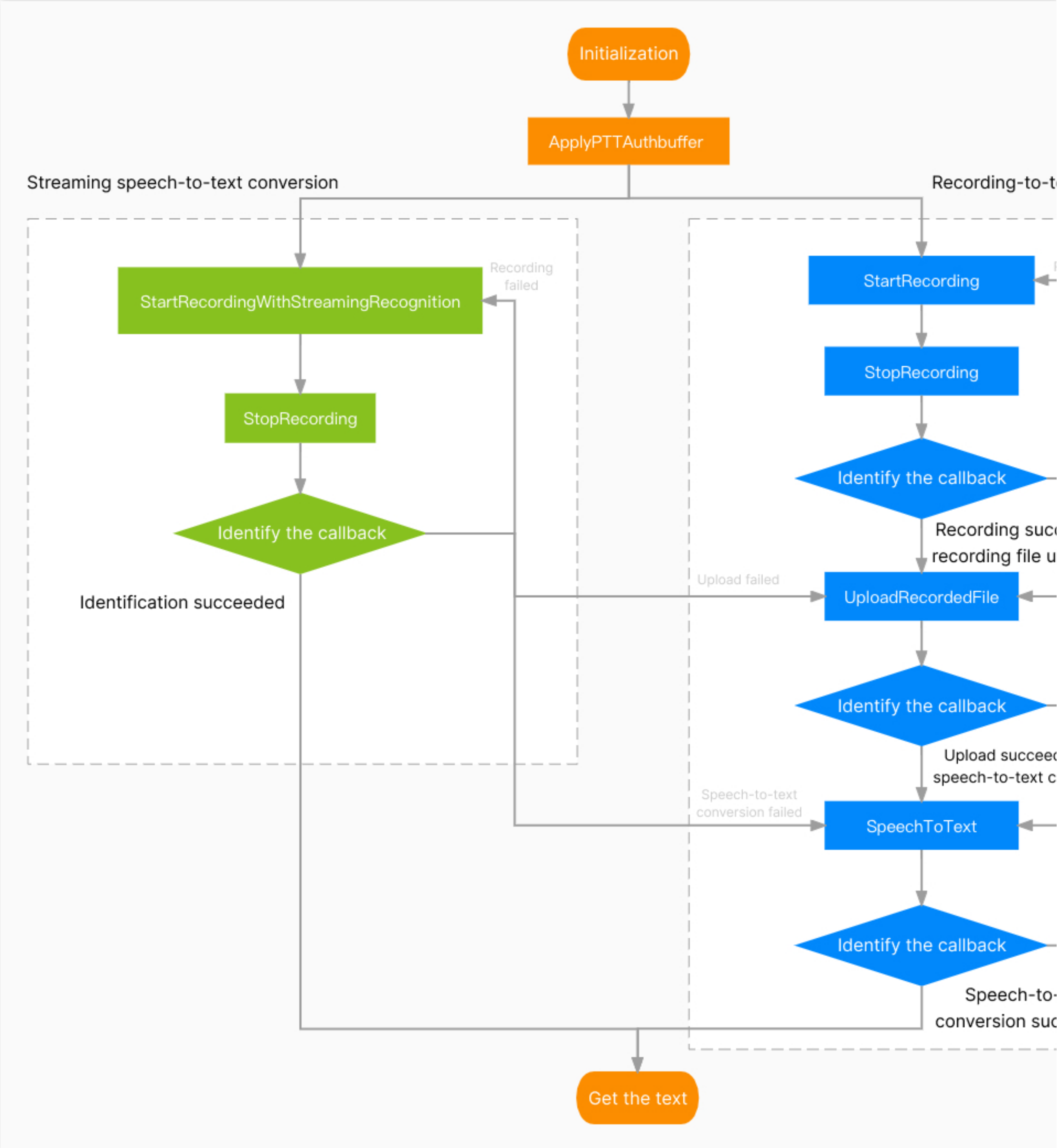
The speech-to-text service consists of fast recording-to-text conversion and streaming speech-to-text conversion.

You do not need to enter a voice chat room when using the voice message service.

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, call the `SetMaxMessageLength` API to set it after initialization.



Flowchart for using the speech-to-text service



API	Description
GenAuthBuffer	Generates the local authentication key
ApplyPTTAuthbuffer	Initializes authentication
SetMaxMessageLength	Specifies the maximum length of voice message

Generating the local authentication key

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.

API prototype

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

Parameter	Type	Description
dwSdkAppID	int	`AppId` from the Tencent Cloud console.
strRoomID	const char*	Enter `null` or an empty string
strOpenID	const char*	User Identifier, which is the same as `openID` during initialization.
strKey	const char*	Permission key from the Tencent Cloud console.
strAuthBuffer	const char*	Returned `authbuff`
bufferLength	int	Length of the `authbuff` passed in. 500 is recommended.

Application authentication

After the authentication information is generated, the authentication is assigned to the SDK.

API prototype

```
ITMGPTT virtual int ApplyPTTAuthbuffer(const char* authBuffer, int authBufferLen)
```

Parameter	Type	Description
authBuffer	const char*	Authentication
authBufferLen	int	Authentication length

Sample code

```
ITMGContextGetInstance()->GetPTT()->ApplyPTTAuthbuffer(authBuffer,authBufferLen);
```

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

API prototype

```
ITMGPTT virtual int SetMaxMessageLength(int msTime)
```

Parameter	Type	Description
msTime	int	Audio duration in ms. Value range: 1000 < msTime <= 58000

Sample code

```
int msTime = 10000;
ITMGContextGetInstance()->GetPTT()->SetMaxMessageLength(msTime);
```

Streaming Speech Recognition

Voice message and speech-to-text APIs

API	Description
StartRecordingWithStreamingRecognition	Starts streaming recording
StopRecording	Stops recording

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call** Stop recording.

API prototype

```
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath)
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath,con
```

Parameter	Type	Description
filePath	const char*	Path of stored audio file
speechLanguage	const char*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translateLanguage	const char*	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContextGetInstance()->GetPTT()->StartRecordingWithStreamingRecognition(filePath
```

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen on callback messages in the `OnEvent` notification, which is as detailed below:

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real-time during the recording, which is equivalent to returning the recognized text while speaking.

The event message will be identified in the `OnEvent` notification based on the actual needs. The passed parameters include the following four messages.

Message Name	Description
result	Return code indicating whether streaming speech recognition is successful
text	Text converted from speech
file_path	Local path of stored recording file
file_id	Backend URL address of recording file, which will be retained for 90 days

Note:

The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Suggested Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.

32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

If the error code 4098 is reported, see FAQs for solutions.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
                switch (eventType) {
                    case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
                    {
                        HandleSTREAM2TEXTComplete(data,true);
                        break;
                        }
                    ...
                            case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_R
                    {
                        HandleSTREAM2TEXTComplete(data, false);
                        break;
                    }
                }
}

void CTMGSDK_For_AudioDlg::HandleSTREAM2TEXTComplete(const char* data, bool isCompl
{
                    std::string strText = "STREAM2TEXT: ret=";
                    strText += data;
                    m_EditMonitor.SetWindowText(MByteToWChar(strText).c_str());
                    Json::Reader reader;
                    Json::Value root;
                    bool parseRet = reader.parse(data, root);
                    if (!parseRet) {
                        ::SetWindowText(m_EditInfo.GetSafeHwnd(),MByteToWChar(std:::
                    }
```



```
                    else
                    {
                        if (isComplete) {
                            ::SetWindowText(m_EditUpload.GetSafeHwnd(), MByteToWCha
                        }
                        else {
                            std::string isruning = "STREAMINGRECOGNITION_IS_RUNNING
                            ::SetWindowText(m_EditUpload.GetSafeHwnd(), MByteToWCha
                        }
                    }
}
```

Voice Message Recording

The recording process is as follows: start recording -> stop recording -> return recording callback -> start the next recording.

Voice message and speech-to-text APIs

API	Description
StartRecording	Starts recording
PauseRecording	Pauses recording
ResumeRecording	Resumes recording
StopRecording	Stops recording
CancelRecording	Cancels recording

Starting recording

This API is used to start recording.

API prototype

```
ITMGPTT virtual int StartRecording(const char* fileDir)
```

Parameter	Type	Description
fileDir	const char*	Path of stored audio file

Sample code

```
char buffer[256]={0};
snprintf(buffer, sizeof(buffer), "%sunreal_ptt_local.file", getFilePath().c_str());
ITMGContextGetInstance()->GetPTT()->StartRecording(buffer);
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

API prototype

```
ITMGPTT virtual int StopRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->StopRecording();
```

Callback for recording start

The recording start result will be returned through the callback.
To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file, which must be accessible and cannot be the `fileid`

Error codes

Error Code Value	Cause	Suggested Solution
4097	Parameter is empty.	Check whether the API parameters in the code are correct.
4098	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.

4099	Recording is in progress.	Ensure that the SDK recording feature is used at the right time.
4100	Audio data is not captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	Mic permission is required for using the SDK. To add the permission, please see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString filepath = JsonObject->GetStringField(TEXT("file_path"));
            std::string path = TCHAR_TO_UTF8(filepath.operator*());
            int duration = 0;
            int filesize = 0;
            if (result == 0) {
                duration = ITMGContextGetInstance()->GetPTT()->GetVoiceFileDuration
                filesize = ITMGContextGetInstance()->GetPTT()->GetFileSize(path.c_s
            }
            onPttRecordFileCompleted(result, filepath, duration, filesize);
        }
    }
}
```

Pausing recording

This API is used to pause recording. If you want to resume recording, please call the `ResumeRecording` API.

API prototype

```
ITMGPTT virtual int PauseRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->PauseRecording();
```

Resuming recording

This API is used to resume recording.

API prototype

```
ITMGPTT virtual int ResumeRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->ResumeRecording();
```

Canceling recording

This API is used to cancel recording. **There is no callback after cancellation**.

API prototype

```
ITMGPTT virtual int CancelRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->CancelRecording();
```

Voice Message Upload, Download, and Playback

API	Description
UploadRecordedFile	Uploads the audio file
DownloadRecordedFile	Downloads the audio file
PlayRecordedFile	Plays back the audio file
StopPlayFile	Stops playing back the audio file
GetFileSize	Gets audio file size

GetVoiceFileDuration	Gets the audio file duration

Uploading an audio file

This API is used to upload an audio file.

API prototype

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of uploaded audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->UploadRecordedFile(filePath);
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file
fileid	FString	File URL path

Error codes

Error Code Value	Cause	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.
8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

8195	A network error occurred.	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString filepath = JsonObject->GetStringField(TEXT("file_path"));
            FString fileid = JsonObject->GetStringField(TEXT("file_id"));
            onPttUploadFileCompleted(result, filepath, fileid);
        }
    }
}
```

Downloading the audio file

This API is used to download an audio file.

API prototype

```
ITMGPTT virtual int DownloadRecordedFile(const char* fileId, const char* filePath)
```

Parameter	Type	Description
fileId	const char*	URL path of file
filePath	const char*	Local path of saved file

Sample code


```
ITMGContextGetInstance()->GetPTT()->DownloadRecordedFile(fileID,filePath);
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file
fileid	FString	URL path of file, which will be retained on the server for 90 days

Error codes

Error Code Value	Cause	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
12291	Network storage system exception.	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	Server file system error.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed during the process of getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.
12295	Failed to decode the packet returned during the process of getting the	Check whether the device can access the internet.

	download parameters.	
12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString filepath = JsonObject->GetStringField(TEXT("file_path"));
            FString fileid = JsonObject->GetStringField(TEXT("file_id"));
            onPttDownloadFileCompleted(result, filepath, fileid);
        }
    }
}
```

Playing back audio

This API is used to play back audio.

API prototype

```
ITMGPTT virtual int PlayRecordedFile(const char* filePath)
ITMGPTT virtual int PlayRecordedFile(const char* filePath, nt voiceType)
```

Parameter	Type	Description
filePath	const char*	Local audio file path
voicetype	int	Voice changer type. For more information, see Voice Changing Effects.

Error codes

Error Code Value	Cause	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
ITMGContextGetInstance()->GetPTT()->PlayRecordedFile(filePath);
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameter includes `result` and `file_path`.

Parameter	Type	Description
code	int	0: playback is completed
filepath	FString	Path of stored recording file

Error codes

Error Code Value	Cause	Suggested Solution
20481	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Parameter is empty.	Check whether the API parameters in the code are correct.
20484	Internal error.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString filepath = JsonObject->GetStringField(TEXT("file_path"));
            onPttPlayFileCompleted(result, filepath);
        }
    }
}
```

Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

API prototype

```
ITMGPTT virtual int StopPlayFile()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->StopPlayFile();
```

Getting audio file size

This API is used to get the size of an audio file.

API prototype

```
ITMGPTT virtual int GetFileSize(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetFileSize(filePath);
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

API prototype

```
ITMGPTT virtual int GetVoiceFileDuration(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetVoiceFileDuration(filePath);
```

Fast Recording-to-Text Conversion

API	Description
SpeechToText	Converts speech to text

Converting audio file to text

This API is used to convert a specified audio file to text.

API prototype

```
ITMGPTT virtual void SpeechToText(const char* fileID)
```

Parameter	Type	Description
fileID	const char*	Audio file URL

Sample code

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(fileID);
```

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

API prototype

```
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage)
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage,cons
```

Parameter	Type	Description
fileID	const char*	The URL of the audio file, which will be retained on the server for 90 days.

speechLanguage	const char*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translatelanguage	const char*	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(filePath,"cmn-Hans-CN","cmn-Hans-C
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result`, `file_path` and `text` (recognized text).

Parameter	Type	Description
result	int32	0: recording is completed
fileid	FString	URL of recording file, which will be retained on the server for 90 days
text	FString	Converted text

Error codes

Error Code Value	Cause	Suggested Solution
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.

32784	Incorrect speech-to-text conversion parameter.	Check whether the API parameter `fileid` in the code is empty.
32785	Speech-to-text translation returned an error.	Error with the backend of voice messaging and speech-to-text feature. Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString text = JsonObject->GetStringField(TEXT("text"));
            FString fileid = JsonObject->GetStringField(TEXT("file_id"));
            onPttSpeech2TextCompleted(result, fileid, text);
        }
    }
}
```

Voice Message Volume Level APIs

API	Description
GetMicLevel	Gets real-time mic volume level
SetMicVolume	Sets recording volume level
GetMicVolume	Gets recording volume level
GetSpeakerLevel	Gets real-time speaker volume
SetSpeakerVolume	Sets playback volume level

GetSpeakerVolume	Gets playback volume level

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetMicLevel()
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetMicLevel();
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

API prototype

```
ITMGPTT virtual int SetMicVolume(int vol)
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->SetMicVolume(100);
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetMicVolume()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetMicVolume();
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetSpeakerLevel()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetSpeakerLevel();
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice message. Value range: 0-200.

API prototype

```
ITMGPTT virtual int SetSpeakerVolume(int vol)
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->SetSpeakerVolume(100);
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice message. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetSpeakerVolume()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetSpeakerVolume();
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for SDK usage analysis.

API prototype

```
ITMGContext virtual const char* GetSDKVersion()
```

Sample code

```
ITMGContextGetInstance()->GetSDKVersion();
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
ITMGContext int SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

`ITMG_LOG_LEVEL` description:

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContextGetInstance()->SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

OS	Path
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API prototype

```
ITMGContext virtual int SetLogPath(const char* logDir)
```

Parameter	Type	Description
logDir	const char*	Path

Sample code

```
cosnt char* logDir = ""// Set a path by yourself
ITMGContext* context = ITMGContextGetInstance();
context->SetLogPath(logDir);
```

Callback Messages

Message	Description	Parameter
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected for network	result; error

	or other reasons	
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	The reconnection to the room started	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	The reconnection to the room succeeded	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status was changed	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker device was changed	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A new speaker device was added	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker device was	result; error

	lost	
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A new mic device was added	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic device was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic device was changed	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room quality changed	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Voice message recording was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Voice message upload was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Voice message download was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Voice message playback was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast speech-to-text conversion was completed	result; text;file_id

ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	Streaming speech-to-text conversion is in progress	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed	result; text;file_id

API 파일

최종 업데이트 날짜: : 2024-01-18 16:05:38

Unreal Engine 개발자가 Tencent Cloud Game Multimedia Engine(GME) 제품 API를 디버깅, 액세스하는 편의성을 위해, Unreal Engine 개발에 적용되는 접속 기술 문서를 소개합니다.

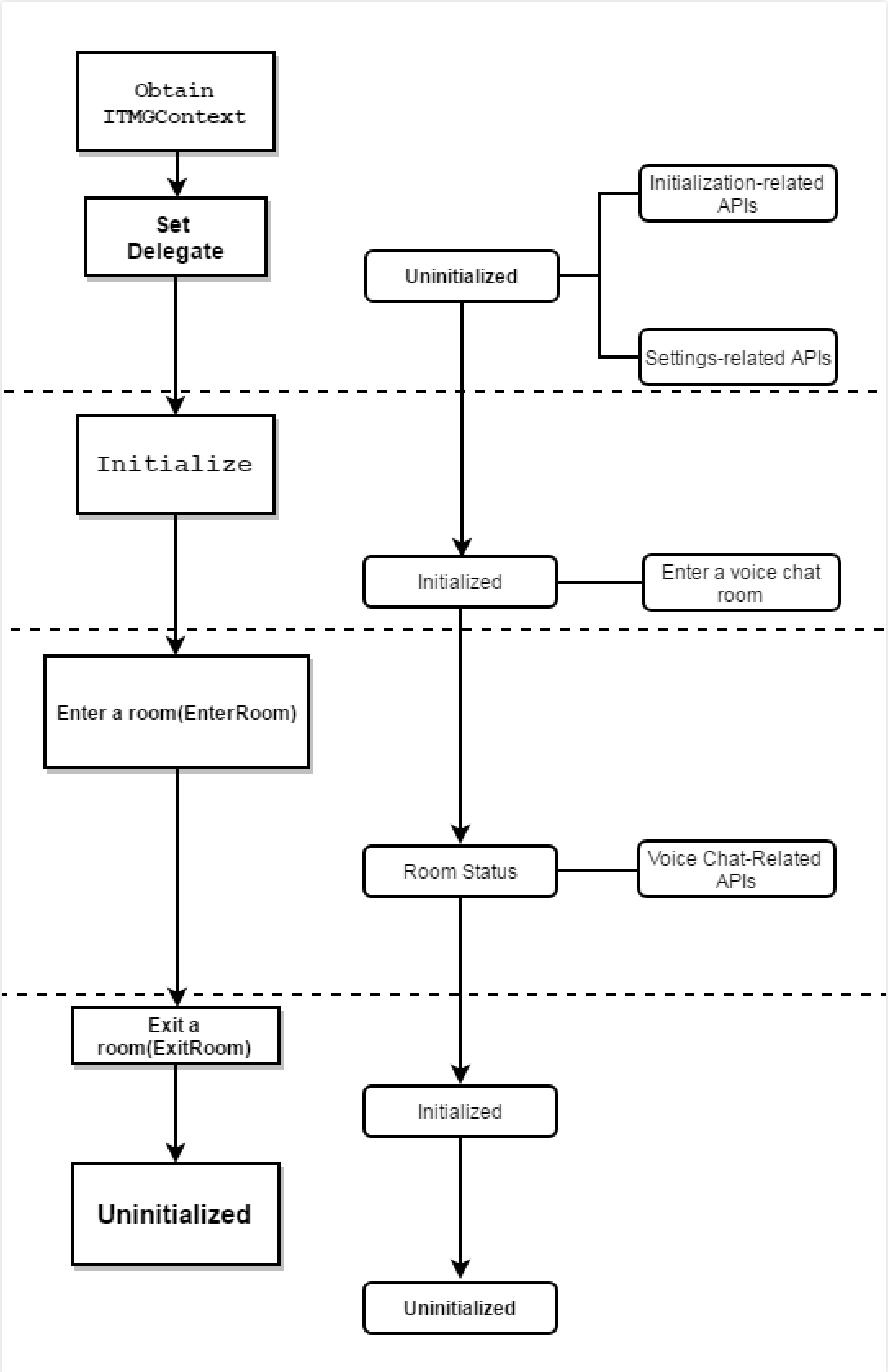
> 이 파일은 GME sdk version : 2.5 맞춤용입니다.

GME 사용 시 주의 중요 사항

중요 인터페이스	인터페이스 의미
Init	GME 초기화
Poll	이벤트 콜백
EnterRoom	방 들어가기
EnableMic	마이크 온
EnableSpeaker	스피커 온

GME 의 사용 전 공정 배치 설정을 하십시오. 그렇지 않는다면 SDK 가 유효하지 않습니다.
GME 의 인터페이스 호출 성공 후의 리턴값은 AV_OK로, 수치는 0입니다.
GME 의 인터페이스 호출은 동일한 스레드에서 사용되어야 합니다.
GME 가 방에 들어가기 위해서는 인증이 필요합니다. 인증 내용 관련 문서를 참조하십시오.
GME 는 주기적으로 Poll 인터페이스 트리거 이벤트 콜백이 필요합니다.
GME 콜백 정보는 콜백 정보 리스트를 참조하십시오.
디바이스의 작업은 방 들어가기 성공 후 가능합니다
에러 코드 상세 내역은 에러코드를 참조하십시오

실시간 음성 메시지 순서도


Obtain ITMGContext
Set Delegate
Initialize
Enter a room(EnterRoom)
Exit a room(ExitRoom)
Uninitialized
Uninitialized
Initialization-related APIs
Settings-related APIs
Initialized
Enter a voice chat room
Room Status
Voice Chat-Related APIs
Initialized
Uninitialized

초기화 관련 인터페이스

초기화하기 전, SDK가 미초기화 단계로 인터페이스 Init를 통해 SDK를 초기화해야 실시간 음성 및 오프라인 음성 메시지를 사용할 수 있습니다.
사용 시 궁금한 사항은 일반 문제를 참조하십시오.

인터페이스	인터페이스 의미
Init	GME 초기화
Poll	이벤트 콜백
Pause	시스템 일시정지
Resume	시스템 복구
Uninit	GME 초기화 취소

준비 과정

GME에 접속하려면 먼저 헤드 파일 tmg_sdk.h를 도입해야 하며, 헤드 파일류는 ITMG Delegate를 상속하여 메시지 전달 및 콜백을 수행합니다.

예시 코드

```
#include "tmg_sdk.h"

class UEDEMO1_API AUEDemoLevelScriptActor : public ALevelScriptActor, public ITMGDe
{
public:
...
private:
...
}
```

단일 항목 설정

EnterRoom 함수 호출에 앞서 ITMGContext를 획득해야 합니다. 모든 호출은 ITMGContext에서 시작하여 ITMGDelegate에서 애플리케이션에 콜백하여야 합니다. 반드시 먼저 설정하십시오.

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->SetTMGDelegate(this);
```

메세지 전송

인터페이스 류는 Delegate 메소드를 사용하여 응용프로그램에 콜백 알림을 보내는데 사용됩니다. 메세지 유형은 ITMG_MAIN_EVENT_TYPE를 참조하세요. data는 Windows 플랫폼에서 json 문자열 형식이며 자세한건key-value 설명 파일을 참조하십시오.

예시 코드

```
//함수 실현:
//UEDemoLevelScriptActor.h:
class UEDEMO1_API AUEDemoLevelScriptActor : public ALevelScriptActor, public SetTMG
{
public:
    void OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data);
}

//UEDemoLevelScriptActor.cpp:
void AUEDemoLevelScriptActor::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* d
    //eventType 판단 및 작업
}
```

SDK 초기화

파라미터 획득과 관련해서 액세스 매뉴얼을 조회하십시오.
이 인터페이스는 텐센트 클라우드 콘솔의 SDKAppID 번호를 파라미터로 사용해야 하며, 여기에 openId를 추가하면 이 openId가 하나의 사용자로 인식됩니다. 규칙은 App 개발자가 자체 작성하고, App 와 중복되지만 않으면 됩니다(현재까지는 INT64만 지원됩니다).

SDK 초기화 후 방 들어가기가 가능합니다

함수 프로토타입

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

파라미터	유형	의미
sdkAppId	char*	텐센트 클라우드 콘솔의 sdkAppId 번호
openId	char*	openId 는 Int64 유형만을 지원합니다（char* 전환 후 입력）. 10000보다 커야 하며, 사용자 식별에 사용됩니다

예시 코드

```
std::string appid = TCHAR_TO_UTF8(CurrentWidget->editAppID->GetText().ToString().op
std::string userId = TCHAR_TO_UTF8(CurrentWidget->editUserID->GetText().ToString().
ITMGContextGetInstance()->Init(appid.c_str(), userId.c_str());
```

이벤트 콜백

Tick에서 주기의 Poll 호출을 통해 이벤트 콜백를 트리거할 수 있습니다.

함수 프로토타입

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```

예시 코드

```
//헤더파일 성명
virtual void Tick(float DeltaSeconds);

//코드 실현
void AUEDemoLevelScriptActor::Tick(float DeltaSeconds)
{
    ITMGContextGetInstance()->Poll();
}
```

시스템 일시정지

시스템에서 pause 이벤트가 발생할 경우, 엔진에 동시에 pause를 공지해야 합니다.

함수 프로토타입

```
ITMGContext int Pause()
```

시스템 복구

시스템에서 Resume 이벤트가 발생할 경우 엔진에 동시에 Resume을 공지해야 합니다. Resume 인터페이스는 실시간 음성 메시지만 복원합니다.

함수 프로토타입

```
ITMGContext int Resume()
```

SDK 초기화 취소

SDK를 초기화 취소하고 미초기화 상태로 들어갑니다. 계정을 전환하려면 초기화 취소가 필요합니다.

함수 프로토타입

```
ITMGContext int Uninit()
```

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->Uninit();
```

실시간 음성 메시지 방 관련 인터페이스

초기화 후, SDK를 호출하여 방으로 들어가야지만 실시간 음성통화가 가능합니다.
궁금한 사항은 실시간 음성 메시지 관련 문제를 참조하십시오.

인터페이스	인터페이스 의미
GenAuthBuffer	초기화 인증
EnterRoom	방 들어가기
IsRoomEntered	방 들어가기 여부
ExitRoom	방 나가기
ChangeRoomType	사용자 방 오디오 유형 수정
GetRoomType	사용자 방 오디오 유형 획득

인증 정보

AuthBuffer 생성은 관련 기능의 암호화와 인증에 사용됩니다. 상세 내역은 인증 보안키를 참조하십시오.오프라인 음성 메시지의 인증 획득 시, 방 번호 파라미터는 반드시 null을 입력해야 합니다.

함수 프로토타입

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

파라미터	유형	의미
dwSdkAppID	int	텐센트 클라우드의 sdkAppId 번호
strRoomID	char*	방 번호는 최대 127자까지 지원됩니다（오프라인 음성 메시지 방 번호 파라미터는 반드시 null을 입력하세요）
strOpenID	char*	사용자 표식
strKey	char*	Tencent Cloud 콘솔 의 보안키
strAuthBuffer	char*	출력된 authbuff
bufferLength	int	전해진 authbuff 길이, 권장 길이 500

예시 코드

```
unsigned int bufferLen = 512;
unsigned char retAuthBuff[512] = {0};
QAVSDK_AuthBuffer_GenAuthBuffer(atoi(SDKAPPID3RD), roomId, "10001", AUTHKEY,retAuth
```

방 추가

생성된 인증 정보로 방에 들어갈 수 있습니다. 콜백하게 될 메시지는 ITMG_MAIN_EVENT_TYPE_ENTER_ROOM입니다. 방 가입 시, 기본값으로 마이크와 스피커는 닫힘 상태입니다. 리턴값이 AV_OK때 성공을 의미합니다.
일반 음성으로 방 들어가기를 진행할 경우, 업무팀에서 범위 음성에 관한 수요가 없는 경우, 일반 방 들어가기 인터페이스를 사용하십시오. 자세한 정보는 범위 음성을 참조하십시오.

함수 프로토타입

```
ITMGContext virtual int EnterRoom(const char*  roomID, ITMG_ROOM_TYPE roomType, con
```

파라미터	유형	의미
roomID	char*	방 번호는 최대 127자까지 지원합니다
roomType	ITMG_ROOM_TYPE	방 오디오 유형
authBuffer	char*	인증 코드
buffLen	int	인증 코드 길이

방 오디오 유형과 관련하여 음질 선택을 참조하십시오.

예시 코드


```
ITMGContext* context = ITMGContextGetInstance();
context->EnterRoom(roomID, ITMG_ROOM_TYPE_STANDARD, (char*)retAuthBuff,bufferLen);
```

방 추가 이벤트 콜백

방 추가 완료 후 발송되는 메시지는 TMG_MAIN_EVENT_TYPE_ENTER_ROOM로 OnEvent 함수에서 판단합니다.

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
    }
}
```

상세 Data

메시지	Data	예제
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}

방에 이미 들어갔는지 여부 판단

이 인터페이스를 호출하여 방에 이미 들어갔는지에 대한 여부를 판단할 수 있습니다. 리턴값은 bool 유형입니다.

함수 프로토타입

```
ITMGContext virtual bool IsRoomEntered()
```

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->IsRoomEntered();
```

방 나가기

이 인터페이스를 호출하여 방 나가기를 할 수 있습니다. 비동기 인터페이스로서, 리턴값이 AV_OK일 때 비동기 딜리버리의 성공을 나타냅니다.

애플리케이션에서 체크 아웃 직후 체크 인이 실행되는 작업 환경이 있는 경우 개발자는 인터페이스 호출 프로세스에서 ExitRoom의 콜백 RoomExitComplete 알림을 기다릴 필요가 없습니다. 인터페이스를 직접 호출하면 됩니다.

함수 프로토타입

```
ITMGContext virtual int ExitRoom()
```

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->ExitRoom();
```

방 나가기 콜백

방 나가기 완료 후 콜백으로, 메시지는 ITMG_MAIN_EVENT_TYPE_EXIT_ROOM입니다.

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
        //처리
        break;
        }
    }
}
```

상세 Data

메시지	Data	예제
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}

사용자 방 오디오 유형 수정

이 인터페이스는 사용자 방 오디오 유형 수정에 사용되며, 결과는 콜백 이벤트를 참조하십시오. 이벤트 유형은 ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE입니다.

함수 프로토타입

```
IITMGContext TMGRoom public void ChangeRoomType((ITMG_ROOM_TYPE roomType)
```

파라미터	유형	의미
roomType	ITMG_ROOM_TYPE	방의 유형을 전환하길 원하는 것으로, 방 오디오 유형은 EnterRoom 인터페이스를 참조하십시오

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
ITMGContextGetInstance()->GetRoom()->ChangeRoomType(ITMG_ROOM_TYPE_FLUENCY);
```

사용자방 오디오 유형 획득

이 인터페이스는 사용자 방의 오디오 유형을 얻는 데 사용되며, 리턴값이 방의 오디오 유형이며, 리턴값이 0인 경우 사용자 방의 오디오 유형을 얻는 데 오류가 발생했음을 의미, 방의 오디오 유형은 EnterRoom 인터페이스를 참조합니다.

함수 프로토타입

```
IITMGContext TMGRoom public  int GetRoomType()
```

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
ITMGContextGetInstance()->GetRoom()->GetRoomType();
```

방 유형 완료 콜백

방 유형 설정 완료 후의 콜백된 이벤트 메시지는 ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE로, 리턴 파라미터는 result, error_info 와 new_room_type입니다. new_room_type 가 의미하는 정보는 아래와 같으며, 이벤트 메시지는 OnEvent 함수에서 판단합니다.

이벤트 하위 유형	파라미터	의미
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	방에 들어오는 동안 가져온 오디오 유형이 방과 일치하지 않음을 나타내는 것으로, 들어간 방의 오디오 유형으로 자동 수정됩니다
ITMG_ROOM_CHANGE_EVENT_START	2	이미 방에 있는 상황에서 오디오 유형을 전환합니다(예를 들어 ChangeRoomType 인터페이스를 호

		출한 후 오디오 유형을 전환합니다)
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	이미 방에 있는 상황에서 오디오 유형 전환 완료를 의미합니다
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	방 내 다른 멤버가 ChangeRoomType 인터페이스를 호출하여 오디오 유형을 변경하고자 하는 경우를 의미합니다

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE == t
        {
        //방 유형 이벤트 처리
      }
}
```

상세 Data

메시지	Data	예제
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; new_room_type	{"error_info":"","new_room_type":0

멤버 상태 변화

이 이벤트는 상태 변화가 있어야 알 수 있으며 상태가 변하지 않는 한 알림이 없습니다. 멤버 상태를 실시간으로 얻으려면 상위에서 알림을 캐시하십시오. 이벤트 메시지는 ITMG_MAIN_EVNET_TYPE_USER_UPDATE로, 그 중 data에는 두가지의 event_id 및 user_list 정보를 포함하고 있습니다. 위 이벤트 메시지는 OnEvent 함수에서 판단합니다.
오디오 이벤트 알림에는 임계 값이 하나 있으며, 이 임계 값을 초과해야 알림이 전송됩니다. 2초 이상 오디오 패킷을 받지 못한 후에야 "멤버가 오디오 패킷 전송을 중지했습니다." 라는 메시지를 알립니다.

event_id	의미	애플리케이션 유지 컨텐츠
ITMG_EVENT_ID_USER_ENTER	멤버가 방에 들어왔습니다	애플리케이션 유지 멤버 리스트
ITMG_EVENT_ID_USER_EXIT	어떤 멤버가 방을 나갔습니다	애플리케이션 유지 멤버 리스트
ITMG_EVENT_ID_USER_HAS_AUDIO	어떤 멤버가 오디오 패킷 발송을 합니다	애플리케이션 유지 통화 멤버 리스트

ITMG_EVENT_ID_USER_NO_AUDIO	어떤 멤버가 오디오 패킷 발송을 중지했습니다	애플리케이션 유지 통화 멤버 리스트

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVNET_TYPE_USER_UPDATE:
        {
        //처리
        //개발자가 파라미터를 분석합니다. 획득한 정보는  eventID 와 user_list 입니다
            switch (eventID)
             {
             case ITMG_EVENT_ID_USER_ENTER:
                  //어떤 멤버가 방에 들어왔습니다
                  break;
             case ITMG_EVENT_ID_USER_EXIT:
                  //어떤 멤버가 방을 나갔습니다
                break;
            case ITMG_EVENT_ID_USER_HAS_AUDIO:
                //어떤 멤버가 오디오 패킷 발송을 합니다
                break;
            case ITMG_EVENT_ID_USER_NO_AUDIO:
                //어떤 멤버가 오디오 패킷 발송을 중지했습니다
                break;
             default:
                break;
            }
        break;
        }
    }
}
```

품질 모니터링 이벤트

품질 모니터링 이벤트 메시지는 ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY로, 리턴 파라미터는 weight, floss 와 delay입니다. 의미하는 정보는 다음과 같으며, OnEvent 함수에서 이벤트 메시지를 판단합니다.

파라미터	의미
weight	범위는 1-5로, 5는 음질 평점이 매우 좋음, 1은 음질 평점이 매우 나쁨으로 거의 사용할 수 없음을 뜻하며, 0은 기본값을 뜻하며 의미가 없다
floss	패킷 손실율

delay	오디오 딜레이 시간（ms）

상세 메시지

메시지	메시지 의미
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	오디오/비디오방 메시지 들어가기
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	오디오/비디오방 메시지 나가기
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	네트워크 등의 이유로 방 메세지 끊김
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	방 유형 변화 이벤트

메시지 관련 상세 Data

메시지	Data	예제
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, pleas your network","result":0}
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; new_room_type	{"error_info":"","new_room_type":0

실시간 음성 채팅 오디오 인터페이스

SDK를 초기화한 후 방에 들어가야 실시간 오디오 음성 메시지 관련 인터페이스를 호출할 수 있습니다.
사용자 인터페이스에서 마이크/스피커 열기/닫기 버튼을 클릭하는 경우 다음과 같은 방식이 권장됩니다:
대부분의 게임류 App에 대해 EnableMic 및 EnableSpeaker 인터페이스 호출을 권장하고 있습니다. 이는 항상 EnableAudioCaptureDevice/EnableAudioSend 와 EnableAudioPlayDevice/EnableAudioRecv 인터페이스를 동시에 사용해야 함을 의미합니다.
다른 유형의 모바일 App의 경우, 예를 들어 소셜 유형 App의 경우, 수집 디바이스를 켜기/끄기 하면 전체 장비(수집 및 재생)가 재시작되며, 이때 App에서 배경 음악이 재생되고 있다면 배경 음악의 재생도 중단됩니다. 업/다운스트림을 제어하는 방식으로 마이크 열기/닫기 를 실행할 경우 재생 디바이스가 중단되지 않습니다. 구체적인 호출 방법은: 방에 들어갈 때 EnableAudioCaptureDevice(true) && EnableAudioPlayDevice(true)를 한 번 호출하고, EnableAudioSend/Recv만을 호출하여 오디오 스트리밍이 송신/수신되는지 여부를 제어합니다.

개별적으로 수집 혹은 재생 디바이스를 릴리즈 하려는 경우, 인터페이스 EnableAudioCaptureDevice 와 EnableAudioPlayDevice를 참조하세요.
pause 를 호출하여 오디오 엔진을 일시정지 합니다. resume 를 호출하여 오디오 엔진을 복구합니다.

인터페이스	인터페이스 의미
GetMicListCount	마이크 갯수 획득
GetMicList	마이크 세기
GetSpeakerListCount	스피커 갯수 획득
GetSpeakerList	스피커 세기
SelectMic	마이크 선정
SelectSpeaker	스피커 선정
EnableMic	마이크 스위치
GetMicState	마이크 상태 획득
EnableAudioCaptureDevice	수집 디바이스 스위치
IsAudioCaptureDeviceEnabled	수집 디바이스 상태 획득
EnableAudioSend	오디오 업스트림 열기/닫기
IsAudioSendEnabled	오디오 업스트림 상태 획득
GetMicLevel	실시간 마이크 사운드 획득
GetSendStreamLevel	오디오 업스트림 실시간 사운드 획득
SetMicVolume	마이크 사운드 설정
GetMicVolume	마이크 사운드 획득
EnableSpeaker	스피커 스위치
GetSpeakerState	스피커 상태 획득
EnableAudioPlayDevice	재생 디바이스 스위치
IsAudioPlayDeviceEnabled	재생 디바이스 상태 획득
EnableAudioRecv	오디오 다운스트림 열기/닫기
IsAudioRecvEnabled	오디오 다운스트림 상태 획득

GetSpeakerLevel	실시간 스피커 사운드 획득
GetRecvStreamLevel	방 내 다른 멤버 다운스트림 실시간 사운드 획득
SetSpeakerVolume	스피커 사운드 설정
GetSpeakerVolume	스피커 사운드 획득
EnableLoopBack	인이어 모니터링 스위치

마이크 갯수 획득

이 인터페이스는 마이크 갯수 획득에 사용됩니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetMicListCount()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicListCount();
```

마이크 갯수 세기

이 인터페이스는 마이크 갯수 세기에 사용됩니다. GetMicListCount 인터페이스와 함께 사용하세요.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetMicList(TMGAudioDeviceInfo* ppDeviceInfoList, int nCou

class TMGAudioDeviceInfo
{
public:
    const char* pDeviceID;
    const char* pDeviceName;
};
```

파라미터	유형	의미
ppDeviceInfoList	TMGAudioDeviceInfo	디바이스 리스트
nCount	int	획득한 마이크 갯수

예시 코드


```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicList(ppDeviceInfoList,nCount);
```

마이크 선택

이 인터페이스는 마이크 선택에 사용됩니다. 만약 디버깅하거나 "DEVICEID_DEFAULT"의 경우 시스템이 기본 디바이스를 선택합니다. 디바이스 ID는 GetMicList 리턴 리스트를 참고하십시오.

함수 프로토타입

```
ITMGAudioCtrl virtual int SelectMic(const char* pMicID)
```

파라미터	유형	의미
pMicID	char*	마이크 디바이스 ID

예시 코드

```
const char* pMicID ="{0.0.1.00000000}.{7b0b712d-3b46-4f7a-bb83-bf9be4047f0d}";
ITMGContextGetInstance()->GetAudioCtrl()->SelectMic(pMicID);
```

마이크 열기/닫기

이 인터페이스는 마이크 열기/닫기 에 사용됩니다. 방 추가 시, 마이크와 스피커 닫힘 상태가 기본값입니다.
EnableMic = EnableAudioCaptureDevice + EnableAudioSend.

함수 프로토타입

```
ITMGAudioCtrl virtual int EnableMic(bool bEnabled)
```

파라미터	유형	의미
bEnabled	bool	마이크를 열기 하여야 하는 경우 인풋 파라미터는 true이고, 마이크를 닫기 하여야 하는 경우, 파라미터는 false 입니다

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableMic(true);
```

마이크 상태 획득

이 인터페이스는 마이크 상태 획득에 사용됩니다. 리턴값 0은 마이크 닫힘 상태를, 리턴값 1은 마이크 열기 상태를 의미합니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetMicState()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicState();
```

수집 디바이스 열기/닫기

이 인터페이스는 수집 디바이스 열기/닫기 에 사용됩니다. 방 추가 시 디바이스 닫기 가 기본값입니다.
방 들어가기 후에만 본 인터페이스 호출이 가능합니다. 방 나가기 후 디바이스가 자동으로 닫힙니다.
모바일의 경우 수집 디바이스 열기 시, 권한 신청이 요구될 수 있습니다. 사운드 유형 조정 등 작업이 필요합니다.

함수 프로토타입

```
ITMGContext virtual int EnableAudioCaptureDevice(bool enable)
```

파라미터	유형	의미
enable	bool	수집 디바이스를 열기 한 경우 파라미터는 true이며, 수집 디바이스 닫기 한 경우 파라미터는 false로 나타납니다

예시 코드

```
수집 디바이스 열기
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioCaptureDevice(true);
```

수집 디바이스 상태 획득

이 인터페이스는 수집 디바이스 상태 획득에 사용됩니다.

함수 프로토타입

```
ITMGContext virtual bool IsAudioCaptureDeviceEnabled()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioCaptureDeviceEnabled();
```

오디오 업스트림 열기/닫기

이 인터페이스는 오디오 업스트림 열기/닫기 에 사용됩니다. 수집 디바이스가 이미 열려 있다면 채취한 오디오 데이터를 보낼 수 있습니다. 수집 디바이스가 닫혀있는 상태면 소리가 나지 않습니다. 수집 디바이스의 열기/닫기 는 인터페이스 EnableAudioCaptureDevice를 참조합니다.

함수 프로토타입

```
ITMGContext  virtual int EnableAudioSend(bool bEnable)
```

파라미터	유형	의미
enable	bool	오디오 업스트림을 열기 하는 경우 파라미터는 true이며, 오디오 업스트림 닫기 를 하면 파라미터는 false로 나타납니다

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioSend(true);
```

오디오 업스트림 상태 획득

이 인터페이스는 오디오 업스트림 상태 획득에 사용됩니다.

함수 프로토타입

```
ITMGContext virtual bool IsAudioSendEnabled()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioSendEnabled();
```

마이크 실시간 사운드 획득

이 인터페이스는 마이크 실시간 사운드 획득에 사용되며, 리턴값은 int 유형입니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetMicLevel()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicLevel();
```

오디오 업스트림 실시간 사운드 획득

이 인터페이스는 오디오 업스트림 실시간 사운드 획득에 사용됩니다. 리턴값은 int 유형이며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetSendStreamLevel()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSendStreamLevel();
```

마이크 사운드 설정

이 인터페이스는 마이크 사운드 설정에 사용됩니다. 파라미터 volume 은 마이크의 볼륨을 설정하는데 사용되며 수치가 0일 때 무음을 표시하고 수치가 100일 때 볼륨이 더이상 증가하지 않음을 의미합니다. 기본값은 100입니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int SetMicVolume(int vol)
```

파라미터	유형	의미
vol	int	사운드 설정, 값 범위는 0 부터 200까지

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->SetMicVolume(vol);
```

마이크 사운드 획득

이 인터페이스는 마이크 사운드 획득에 사용됩니다. 리턴값은 int 유형이며, 리턴값 101은 SetMicVolume 인터페이스가 호출되지 않음을 의미합니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetMicVolume()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicVolume();
```

스피커 갯수 획득

이 인터페이스는 스피커 갯수 획득에 사용됩니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetSpeakerListCount()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerListCount();
```

스피커 갯수 세기

이 인터페이스는 스피커 갯수 세기에 사용됩니다. GetSpeakerListCount 인터페이스와 함께 사용하세요.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetSpeakerList(TMGAudioDeviceInfo* ppDeviceInfoList, int

class TMGAudioDeviceInfo
{
public:
    const char* pDeviceID;
    const char* pDeviceName;
};
```

파라미터	유형	의미
ppDeviceInfoList	TMGAudioDeviceInfo	디바이스 리스트
nCount	int	획득한 스피커 갯수

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerList(ppDeviceInfoList,nCount);
```

선택한 스피커

이 인터페이스는 재생할 장치를 선택하는 데 사용됩니다. 디버깅하거나 "DEVICEID_DEFAULT"가 전송될 경우, 시스템이 기본값 재생 디바이스를 선택합니다. 디바이스 ID는 GetSpeakerList 리턴 리스트에서 가져옵니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int SelectSpeaker(const char* pSpeakerID)
```

파라미터	유형	의미
pSpeakerID	char*	스피커 디바이스 ID

예시 코드

```
const char* pSpeakerID ="{0.0.1.00000000}.{7b0b712d-3b46-4f7a-bb83-bf9be4047f0d}";
ITMGContextGetInstance()->GetAudioCtrl()->SelectSpeaker(pSpeakerID);
```

스피커 열기/닫기

이 인터페이스는 스피커 열기/닫기 에 사용됩니다.
EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv.

함수 프로토타입

```
ITMGAudioCtrl virtual int EnableSpeaker(bool enabled)
```

파라미터	유형	의미
enable	bool	스피커를 열기 하는 경우 파라미터는 false이고, 스피커를 닫기 하면 true 가 나타납니다

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableSpeaker(true);
```

스피커 상태 획득

이 인터페이스는 스피커 상태 획득에 사용됩니다. 리턴값 0은 스피커 닫기 상태를 의미하고, 리턴값 1은 스피커 열기 상태를 의미합니다. 리턴값 2는 스피커 디바이스가 현재 가동 중을 의미합니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetSpeakerState()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerState();
```

재생 디바이스 열기/닫기

이 인터페이스는 재생 디바이스 열기/닫기 에 사용됩니다.

함수 프로토타입

```
ITMGContext virtual int EnableAudioPlayDevice(bool enable)
```

파라미터	유형	의미
enable	bool	재생 디바이스를 꺼야 하는 경우 파라미터는 false이고, 재생 디바이스를 켜면 true 가 나타납니다

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioPlayDevice(true);
```

재생 디바이스 상태 획득

이 인터페이스는 재생 디바이스 상태 획득에 사용됩니다.

함수 프로토타입

```
ITMGContext virtual bool IsAudioPlayDeviceEnabled()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioPlayDeviceEnabled();
```

오디오 다운스트림 켜기/끄기

이 인터페이스는 오디오 다운스트림 켜기/끄기 에 사용됩니다. 재생 장치가 이미 열려 있다면 방에 있는 다른 사람의 오디오 데이터를 재생합니다. 재생 장치가 켜져 있지 않으면 소리가 나지 않습니다. 재생 장치의 켜기/ 끄기는 인터페이스 EnableAudioPlayDevice를 참조합니다.

함수 프로토타입

```
ITMGContext virtual int EnableAudioRecv(bool enable)
```

파라미터	유형	의미
enable	bool	오디오 다운스트림을 켜야 하는 경우 파라미터는 true이고, 오디오 다운스트림을 끄면 false가 나타납니다

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioRecv(true);
```

오디오 다운스트림 상태 획득

이 인터페이스는 오디오 다운스트림 상태 획득에 사용됩니다.

함수 프로토타입

```
ITMGContext virtual bool IsAudioRecvEnabled()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioRecvEnabled();
```

스피커 실시간 사운드 획득

이 인터페이스는 스피커 실시간 사운드 획득에 사용됩니다. 리턴값은 int 유형이며, 스피커 실시간 사운드를 의미합니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetSpeakerLevel()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerLevel();
```

방 내 다른 멤버 다운스트림 실시간 사운드 획득

이 인터페이스는 방 내 다른 멤버 다운스트림 실시간 사운드 획득에 사용됩니다. 리턴값은 int 유형으며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetRecvStreamLevel(const char* openId)
```

파라미터	유형	의미
openId	char*	방 내 다른 멤버의 openId

예시 코드

```
iter->second.level = ITMGContextGetInstance()->GetAudioCtrl()->GetRecvStreamLevel(i
```

스피커 사운드 설정

이 인터페이스는 스피커 사운드 설정에 사용됩니다.
파라미터 volume 은 스피커 볼륨을 설정하는데 사용되며 수치가 0일 때 무음을 표시하고 수치가 100일 때 볼륨이 더 이상 증가하지 않음을 나타냅니다. 기본값은 100입니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int SetSpeakerVolume(int vol)
```

파라미터	유형	의미
vol	int	사운드 설정, 범위 값 0부터 200까지

예시 코드

```
int vol = 100;
ITMGContextGetInstance()->GetAudioCtrl()->SetSpeakerVolume(vol);
```

스피커 사운드 획득

이 인터페이스는 스피커 사운드 획득에 사용됩니다. 리턴값은 int유형이며, 스피커의 사운드를 의미합니다. 리턴값 101은 인터페이스 SetSpeakerVolume를 호출한 적이 없음을 의미합니다.
Level은 실시간 볼륨을 뜻하고, Volume은 스피커 사운드를 의미하며, 최종 사운드는 Level*Volume%에 해당합니다. 예를 들어: 실시간 볼륨이 수치가 100, Volume의 수치가 60일 때, 결과적으로 발생하는 소리의 수치는 60임을 뜻합니다.

함수 프로토타입

```
ITMGAudioCtrl virtual int GetSpeakerVolume()
```

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerVolume();
```

인이어 모니터링 작동

이 인터페이스는 인이어 모니터링 작동에 사용됩니다.

함수 프로토타입

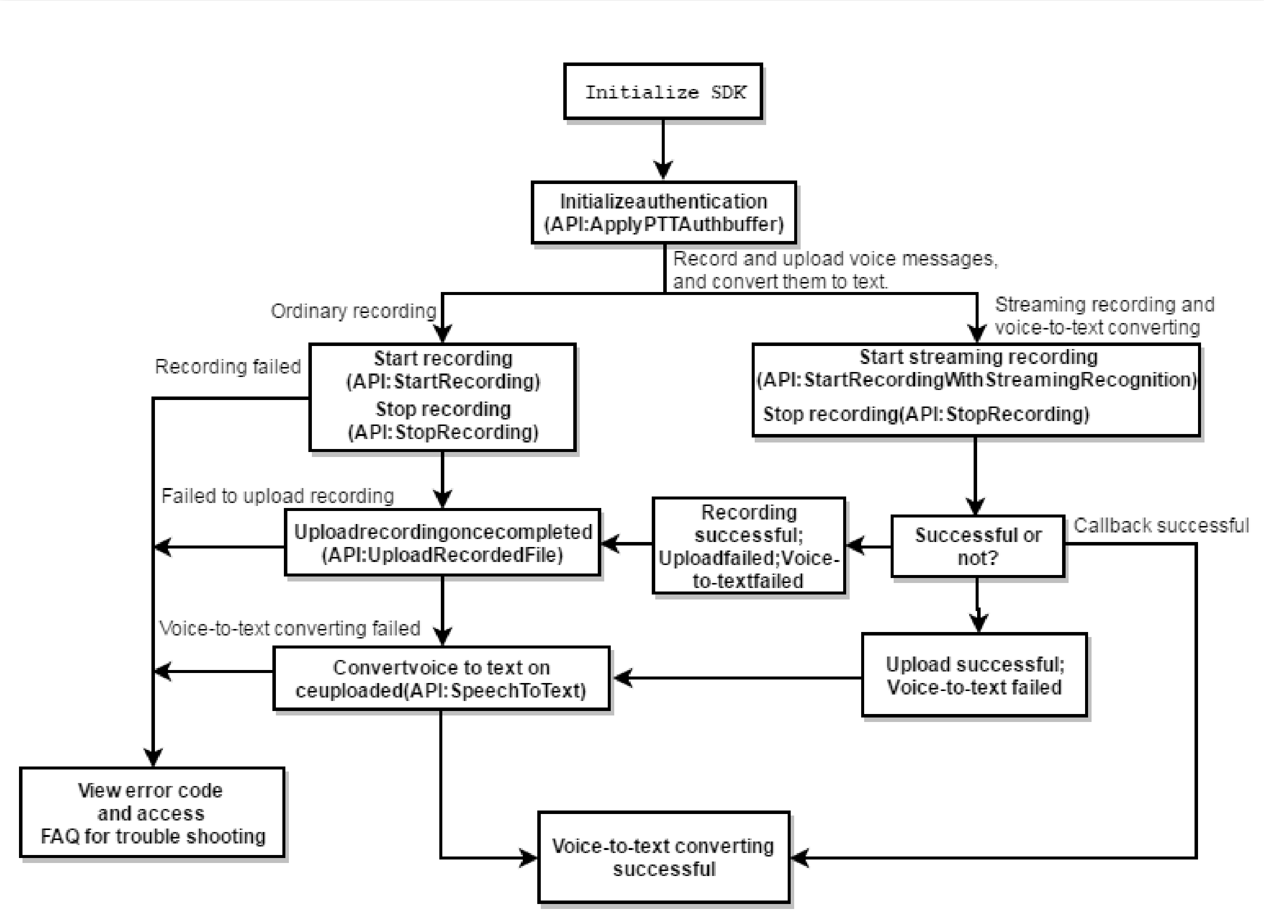
```
ITMGAudioCtrl virtual int EnableLoopBack(bool enable)
```

파라미터	유형	의미
enable	bool	설정이 운행 가능한지 여부

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableLoopBack(true);
```

오프라인 음성 메시지 음성 문자 간 변환 순서도



오프라인 음성 메시지

초기화하기 전에, SDK가 초기화되지 않은 단계로 인터페이스 Init를 통해 SDK를 초기화해야 실시간 음성 및 오프라인 음성 메시지를 사용할 수 있습니다.
궁금한 사항은[오프라인 음성 메시지 관련 문제]를 참조하십시오
(https://www.tencentcloud.com/document/product/607/30258).

초기화 관련 인터페이스

인터페이스	인터페이스 의미
Init	GME 초기화
Poll	이벤트 콜백
Pause	시스템 일시정지
Resume	시스템 복구

Uninit	GME 초기화 취소

오프라인 음성 메시지 관련 인터페이스

인터페이스	인터페이스 의미
ApplyPTTAuthbuffer	인증 초기화
SetMaxMessageLength	음성 메시지 최대 길이 제한
StartRecording	녹음 시작
StartRecordingWithStreamingRecognition	스트리밍 녹음 시작
PauseRecording	녹음 일시정지
ResumeRecording	녹음 복구
StopRecording	녹음 중지
CancelRecording	녹음 취소
GetMicLevel	오프라인 음성 메시지 실시간 마이크 사운드 획득
SetMicVolume	오프라인 음성 메시지 녹화 사운드 설정
GetMicVolume	오프라인 음성 메시지 녹화 사운드 획득
GetSpeakerLevel	오프라인 음성 메시지 실시간 스피커 사운드 획득
SetSpeakerVolume	오프라인 음성 메시지 재생 사운드 설정
GetSpeakerVolume	오프라인 음성 메시지 재생 사운드 획득
UploadRecordedFile	음성 파일 업로드
DownloadRecordedFile	음성 파일 다운로드
PlayRecordedFile	음성 파일 재생
StopPlayFile	음성 파일 재생 중지
GetFileSize	음성 파일 크기
GetVoiceFileDuration	음성 파일 길이
SpeechToText	음성을 문자로 인식

인증 초기화

SDK를 초기화한 후 인증 초기화를 호출합니다. authBuffer 획득은 윗글의 실시간 음성 메시지 인증 정보 인터페이스를 참조합니다.

함수 프로토타입

```
ITMGPTT virtual int ApplyPTTAuthbuffer(const char* authBuffer, int authBufferLen)
```

파라미터	유형	의미
authBuffer	char*	인증
authBufferLen	int	인증 길이

예시 코드

```
ITMGContextGetInstance()->GetPTT()->ApplyPTTAuthbuffer(authBuffer,authBufferLen);
```

음성 메시지 최대 길이 제한

음성 메시지 최대 길이를 60초까지로 제한합니다.

함수 프로토타입

```
ITMGPTT virtual int SetMaxMessageLength(int msTime)
```

파라미터	유형	의미
msTime	int	음성 시간, 단위 ms

예시 코드

```
int msTime = 10;
ITMGContextGetInstance()->GetPTT()->SetMaxMessageLength(msTime);
```

녹음 시작

이 인터페이스는 녹음 시작에 사용됩니다. 녹음 파일 업로드 완료 후에만 음성 문자 간 변환 등 작업을 진행할 수 있습니다.

함수 프로토타입

```
ITMGPTT virtual int StartRecording(const char* fileDir)
```

파라미터	유형	의미
fileDir	char*	보관된 음성 파일 경로

예시 코드

```
ITMGContextGetInstance()->GetPTT()->StartRecording(fileDir);
```

녹음 시작 콜백

녹음 시작 완료 후의 콜백은 함수 OnEvent를 호출합니다. 이벤트 메시지는 ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE로, OnEvent 함수에서 판단합니다.
전송된 파라미터에는 result 와 file_path, 이 두 가지 정보가 포함되어 있습니다.

에러 코드

에러 코드	원인	권장 방안
4097	파라미터 비어 있음	코드 중 인터페이스 파라미터가 올바른지 점검합니다
4098	초기화 오류	디바이스가 점용되었는지, 권한이 올바른지, 초기화가 정상적인지 점검합니다
4099	녹화 중	SDK 녹화 기능이 정확한 타이밍에 사용되는지 확인합니다
4100	가청 주파수 데이터가 수집 안 됨	마이크 디바이스가 정상인지 점검합니다
4101	녹음 시, 파일 액세스 오류	파일이 존재하는지, 파일 경로가 합법적인지 확인합니다
4102	마이크 권한 미부여 오류	SDK 를 사용하기 위해서는 마이크 권한이 필요합니다. 권한 추가는 엔진이나 플랫폼의 SDK 공정 배치 문서를 참조하십시오
4103	녹음 시간이 너무 짧음으로 오류	우선, 녹음 시간 단위는 ms로 제한하고 있습니다. 파라미터가 올바른지 점검하세요. 그리고 녹음 시간은 1000ms 이상이어야 녹음 성공으로 간주합니다
4104	녹음 작업 운행 안됨	이미 녹음 운행 인터페이스가 호출된 것은 아닌지 점검합니다

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
        {
        //처리
        break;
        }
    }
}
```

스트리밍 음성 인식 가동

이 인터페이스는 스트리밍 음성 인식을 작동시키는 데 사용되며, 동시에 콜백 중에 실시간 음성 문자간 변환을 할 수 있습니다. 언어를 지정하여 인식할 수 있으며, 음성에서 인식된 정보를 지정된 언어로 번역하여 출력할 수 있습니다.

함수 프로토타입

```
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath)
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath,con
```

파라미터	유형	의미
filePath	char*	보관된 파일 경로
speechLanguage	char*	지정 문자로 인식된 언어 파라미터와 관련하여 음성 문자 변환된 언어 파라미터 참고 리스트를 참조하십시오
translateLanguage	char*	지정 문자로 번역된 언어 파라미터와 관련하여 음성 문자 변환된 언어 파라미터 참고 리스트를 참조하십시오（이 파라미터는 일시적으로 사용할 수 없으므로 speechLanguage 와 같은 파라미터로 작성하십시오)

예시 코드

```
ITMGContextGetInstance()->GetPTT()->StartRecordingWithStreamingRecognition(filePath
```

스트리밍 음성 인식 가동 콜백

스트리밍 음성 인식 가동 완료된 후의 콜백은 함수 OnEvent를 호출합니다. 이벤트 메시지는 ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE로, OnEvent 함수에서 판단합니다. 전달된 파라미터에는 다음과 같은 네 가지 정보가 포함되어 있습니다.

메시지 이름	의미
result	스트리밍 음성 인식이 성공적으로 출력되었는지 여부를 판단하는 데 사용됨
text	음성 문자 변환 인식의 텍스트
file_path	녹음이 보관된 로컬 주소
file_id	녹음의 백그라운드 url 주소

예시 코드

에러 코드	의미	처리 방법
32775	스트리밍 음성 문자 변환에 실패하였으나, 녹음엔 성공함	UploadRecordedFile 인터페이스를 호출하여 녹음 파일을 업로드 한 뒤, SpeechToText 인터페이스를 호출하여 음성 문자 변환 작업을 진행하십시오
32777	스트리밍 음성 문자 변환에 실패하였으나, 녹음과 업로드에 성공함	리턴 메시지의 업로드 성공 백그라운드 url 주소로, SpeechToText 인터페이스를 호출하여 음성 문자 변환 작업을 진행하십시오
32786	스트리밍 음성 문자 변환 실패	스트리밍 녹화 상태에서, 스트리밍 녹화 인터페이스 실행 결과가 출력될 때까지 기다리십시오

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
        {
        //처리
        break;
        }
    }
```

```
}
```

녹음 일시중지

이 인터페이스는 녹음 일시중지에 사용됩니다. 녹음을 복구하시려면 ResumeRecording 인터페이스를 호출하세요.

함수 프로토타입

```
ITMGPTT virtual int PauseRecording()
```

예시 코드

```
ITMGContextGetInstance()->GetPTT()->PauseRecording();
```

녹음 복구

이 인터페이스는 녹음 복구에 사용됩니다.

함수 프로토타입

```
ITMGPTT virtual int ResumeRecording()
```

예시 코드

```
ITMGContextGetInstance()->GetPTT()->ResumeRecording();
```

녹음 중지

이 인터페이스는 녹음을 중지하는 데 사용됩니다. 이 인터페이스는 비동기 인터페이스이며, 녹음을 중지하면 녹음 완료 콜백이 생성되고, 성공한 이후에 녹음 파일 사용이 가능합니다.

함수 프로토타입

```
ITMGPTT virtual int StopRecording()
```

예시 코드

```
ITMGContextGetInstance()->GetPTT()->StopRecording();
```

녹음 취소

이 인터페이스를 호출하여 녹음을 취소합니다. 취소 후에는 콜백이 없습니다.

함수 프로토타입

```
ITMGPTT virtual int CancelRecording()
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

오프라인 음성 메시지 마이크 실시간 사운드 획득

이 인터페이스는 오프라인 음성 메시지 마이크 실시간 사운드 획득에 사용됩니다. 리턴값은 int유형이며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

오프라인 음성 메시지 녹화 사운드 설정

이 인터페이스는 오프라인 음성 메시지 녹화 사운드 설정에 사용되며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGPTT virtual int SetMicVolume(int vol)
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

오프라인 음성 메시지 녹화 사운드 획득

이 인터페이스는 오프라인 음성 메시지 녹화 사운드 획득에 사용됩니다. 리턴값은 int유형이며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

스피커 실시간 사운드 획득

이 인터페이스는 스피커 실시간 사운드 획득에 사용됩니다. 리턴값은 int유형이며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

오프라인 음성 메시지 재생 사운드 설정

이 인터페이스는 오프라인 음성 메시지 재생 사운드 설정에 사용되며, 값 범위는 0부터 100까지입니다.

함수 프로토타입

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

오프라인 음성 메시지 재생 사운드 획득

이 인터페이스는 오프라인 음성 메시지 재생 사운드 획득에 사용됩니다. 리턴값은 int 유형으로, 값 범위는 0 부터 100 까지입니다.

함수 프로토타입

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

예시 코드

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

음성 파일 업로드

이 인터페이스는 음성 파일 업로드에 사용됩니다.

함수 프로토타입

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

파라미터	유형	의미
filePath	char*	파일 업로드 경로

예시 코드

```
ITMGContextGetInstance()->GetPTT()->UploadRecordedFile(filePath);
```

음성 파일 업로드 완료 콜백

음성 파일 업로드 완료 후, 이벤트 소식은 ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE로, OnEvent 함수에서 판단합니다.
전송된 파라미터에는 result, file_path 와 file_id, 이 세가지 정보가 포함되어 있습니다.

에러 코드

에러 코드	원인	권장 방안
8193	파일 업로드 시, 파일 액세스 오류	파일이 존재하는지, 파일 경로가 합법적인지 확인합니다
8194	서명 교정 실패 오류	인증 보안키가 올바른지, 초기화 오프라인 음성 메시지가 있는지 점검합니다
8195	네트워크 오류	디바이스 네트워크가 외부 네트워크 환경에 정상적으로 방문할 수 있는지 점검합니다
8196	업로드된 파라미터 획득 과정에서 네트워크 오류	인증이 올바른지, 디바이스 네트워크가 외부 네트워크 환경에 정상적으로 방문할 수 있는지 점검합니다
8197	업로드된 파라미터 획득 과정에서 롤백 데이터 비어있음	인증이 올바른지, 디바이스 네트워크가 외부 네트워크 환경에 정상적으로 방문할 수 있는지 점검합니다
8198	업로드된 파라미터 획득 과정에서 롤백 데이터 압축해제 실패	인증이 올바른지, 디바이스 네트워크가 외부 네트워크 환경에 정상적으로 방문할 수 있는지 점검합니다
8200	appinfo 설정 안됨	apply 인터페이스가 호출되어 있는지, 인풋 파라미터가 비어 있는지 점검합니다

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
        {
        //처리
        break;
        }
    }
}
```

음성 파일 다운로드

이 인터페이스는 음성 파일 다운로드에 사용됩니다.

함수 프로토타입

```
ITMGPTT virtual int DownloadRecordedFile(const char* fileId, const char* filePath)
```

파라미터	유형	의미
fileId	char*	파일 url 경로
filePath	char*	파일 로컬 저장 경로

예시 코드

```
ITMGContextGetInstance()->GetPTT()->DownloadRecordedFile(fileID,filePath);
```

음성 파일 다운로드 완료 콜백

음성 파일 다운로드 후, 이벤트 메시지는 ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE로, OnEvent 함수에서 판단합니다.

에러 코드

에러 코드	원인	권장 방안

12289	파일 다운로드 시, 파일 엑세스 오류	파일 경로가 합법적인지 점검합니다
12290	서명 교정 실패	인증 보안키가 올바른지, 초기화 오프라인 음성 메시지가 있는지 점검합니다
12291	네트워크 스토리지 시스템 이상	서버에서 음성 파일 획득에 실패했습니다. 인터페이스 파라미터 fileid 가 올바른지, 네트워크가 정상인지, cos 파일이 존재하는지 점검합니다
12292	서버 파일 시스템 오류	디바이스 네트워크가 정상적으로 외부 네트워크 환경에 방문할 수 있는지, 서버에 파일이 존재하는지 점검합니다
12293	다운로드된 파라미터 획득 과정에서, HTTP 네트워크 오류	디바이스 네트워크가 정상적으로 외부 네트워크 환경에 방문할 수 있는지 점검합니다
12294	다운로드된 파라미터 획득 과정에서, 롤백 데이터 비어있음	디바이스 네트워크가 정상적으로 외부 네트워크 환경에 방문할 수 있는지 점검합니다
12295	다운로드된 파라미터 획득 과정에서, 롤백 데이터 압축해제 실패	디바이스 네트워크가 정상적으로 외부 네트워크 환경에 방문할 수 있는지 점검합니다
12297	appinfo 설정 안됨	인증 보안키가 올바른지, 초기화 오프라인 음성 메시지가 있는지 점검합니다

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE:
        {
        //처리
        break;
        }
    }
}
```


음성 재생

이 인터페이스는 음성 재생에 사용됩니다.

함수 프로토타입

```
ITMGPTT virtual int PlayRecordedFile(const char* filePath)
```

파라미터	유형	의미
filePath	char*	파일 경로

에러 코드

에러코드	원인	권장 방안
20485	재생 미시작	파일이 존재하는지, 파일 경로가 합법적인지 확인합니다

예시 코드

```
ITMGContextGetInstance()->GetPTT()->PlayRecordedFile(filePath);
```

음성 재생 콜백

음성 재생 콜백은, 이벤트 메시지 ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE로, OnEvent 함수에서 판단합니다.
전송된 파라미터는 result와 file_path, 이 두가지 정보를 포함하고 있습니다.

에러 코드

에러 코드	원인	권장 방안
20481	초기화 오류	디바이스가 점용되었는지, 권한이 정상인지, 초기화가 정상적인지 점검합니다
20482	재생 중인 상황에서, 중도 중단하고 다음 재생 시도하였으나 실패함(정상적으로는 중단이 가능함)	코드 로직이 올바른지 점검합니다
20483	파라미터 비어있음	코드 중 인터페이스 파라미터가 정확한지 점검합니다
20484	내부 오류	초기화 재생 오류로, 암호 해독 실패 등의 문제로 이 에러 코드가 생성됩니다. 로그를 위치를 분석하여 문제를 해결합니다

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE:
        {
        //처리
        break;
        }
    }
}
```

음성 재생 중지

이 인터페이스는 음성 재생 중지에 사용됩니다. 음성 재생 중지하여도 재생 완료로 콜백합니다.

함수 프로토타입

```
ITMGPTT virtual int StopPlayFile()
```

예시 코드

```
ITMGContextGetInstance()->GetPTT()->StopPlayFile();
```

음성 파일 크기 획득

이 인터페이스를 통해 음성 파일의 크기를 획득합니다.

함수 프로토타입

```
ITMGPTT virtual int GetFileSize(const char* filePath)
```

파라미터	유형	의미
filePath	char*	음성 파일의 경로

예시 코드

```
ITMGContextGetInstance()->GetPTT()->GetFileSize(filePath);
```

음성 파일 길이 획득

이 인터페이스는 음성 파일의 길이 획득에 사용되며, 단위는 ms입니다.

함수 프로토타입

```
ITMGPTT virtual int GetVoiceFileDuration(const char* filePath)
```

파라미터	유형	의미
filePath	char*	음성 파일의 경로

예시 코드

```
ITMGContextGetInstance()->GetPTT()->GetVoiceFileDuration(filePath);
```

지정 음성 파일을 문자로 인식

이 인터페이스는 지정 음성 파일을 문자로 인식합니다.

함수 프로토타입

```
ITMGPTT virtual void SpeechToText(const char* fileID)
```

파라미터	유형	의미
fileID	char*	음성 파일 url

예시 코드

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(fileID);
```

지정 음성 파일을 문자로 번역(지정 언어)

이 인터페이스는 언어를 지정하여 인식할 수도 있고 음성에서 인식된 정보를 지정된 언어로 번역하여 출력할 수도 있습니다.

함수 프로토타입

```
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage,cons
```

파라미터	유형	의미
fileID	char*	음성 파일 url
speechLanguage	char*	지정 문자로 인식된 음성 파라미터와 관련하여음성 문자 변환된 음성 파라미터 참고 리스트를 참조하십시오
translatelanguage	char*	지정 문자로 반역된 음성 파라미터와 관련하여음성 문자 변환된 음성 파라미터 참고 리스트를 참조하십시오（이 파라미터는 일시적으로 유효하지 않으므로, 인풋 파라미터는 speechLanguage와 일치해야 합니다）

예시 코드

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(filePath,"cmn-Hans-CN","cmn-Hans-C
```

인식 콜백

지정한 음성 파일을 문자로 인식한 콜백은, 이벤트 메시지 ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE이며, OnEvent 함수에서 판단합니다.
전송된 파라미터에는 result, file_path 와 text, 세가지 정보가 포함되어 있습니다. 그 중 text 가 인식된 문자입니다.

에러 코드

에러 코드 값	원인	권장방안
32769	내부 오류	로그를 분석하여 백그라운드에서 클라이언트에게 출력되는 진정한 에러 코드를 얻고 뒤 백그라운드 동료에게 연락하여 해결을 협조하십시오.
32770	네트워크 오류	디바이스 네트워크가 외부 네트워크 환경에 정상적으로 방문할 수 있는지 점검합니다
32772	롤백 압축해제 실패	로그를 분석하여 백그라운드에서 클라이언트에게 출력되는 진정한 에러 코드를 얻고 뒤 백그라운드 동료에게 연락하여 해결을 협조하십시오.
32774	appinfo 설정 안됨	인증 보안키가 올바른지, 초기화 오프라인 음성 메시지가 있는지 점검합니다
32776	authbuffer 교정 실패	authbuffer 가 올바른지 확인합니다
32784	음성 문자 변환 파라미터 오류	코드 중 인터페이스 파라미터 fileid 가 비어있는지 점검하세요

32785	음성을 번역 텍스트로 변환 출력 오류	오프라인 음성 메시지 백그라운드 오류의 경우, 로그를 분석하여 백그라운드에서 클라이언트에게 출력되는 진정한 에러 코드를 얻고 뒤 백그라운드 동료에게 연락하여 해결을 협조하십시오

예시 코드

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        //처리
        break;
        }
        ...
        case ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
        {
        //처리
        break;
        }
    }
}
```

고급 API

진단 정보 획득

오디오 비디오 통화의 실시간 통화 품질에 대한 정보를 획득합니다. 이 인터페이스는 주로 실시간 통화 품질, 배찰 문제 등을 살펴보는 데 사용되며 업무팀에서는 무시하셔도 무관합니다.

함수 프로토타입

```
ITMGRoom virtual const char* GetQualityTips()
```

예시 코드

```
ITMGContextGetInstance()->GetRoom()->GetQualityTips();
```

버전 번호 획득

함수 프로토타입

```
ITMGContext virtual const char* GetSDKVersion()
```

예시 코드

```
ITMGContextGetInstance()->GetSDKVersion();
```

로그 레벨 인쇄 설정

로그 레벨 인쇄 설정에 사용됩니다. 기본값 레벨 유지를 권장합니다.

함수 프로토타입

```
ITMGContext int SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

파라미터 의미

파라미터	유형	의미
levelWrite	ITMG_LOG_LEVEL	입력 로그 레벨을 설정하십시오. TMG_LOG_LEVEL_NONE은 사용하지 않음을 의미합니다.
levelPrint	ITMG_LOG_LEVEL	인쇄 로그 레벨을 설정하십시오. TMG_LOG_LEVEL_NONE은 인쇄하지 않음을 의미합니다.

ITMG_LOG_LEVEL	내용
TMG_LOG_LEVEL_NONE=0	로그를 인쇄하지 않습니다.
TMG_LOG_LEVEL_ERROR=1	로그 인쇄 에러코드(기본값)
TMG_LOG_LEVEL_INFO=2	로그 표시 인쇄
TMG_LOG_LEVEL_DEBUG=3	개발 디버깅 로그 인쇄
TMG_LOG_LEVEL_VERBOSE=4	고빈도 로그 인쇄

예시 코드

```
ITMGContext* context = ITMGContextGetInstance();
context->SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

로그 루트 출력 설정

로그 루트 출력 설정에 사용됩니다.
기본 경로는 :

플랫폼	경로
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

함수 프로토타입

```
ITMGContext virtual int SetLogPath(const char* logDir)
```

파라미터	유형	의미
logDir	char*	경로

예시 코드

```
cosnt char* logDir = ""//자체 설정 경로

ITMGContext* context = ITMGContextGetInstance();
context->SetLogPath(logDir);
```

가청주파수 데이터 블랙리스트에 추가

임의 ID를 가청주파수 데이터 블랙리스트에 추가합니다. 리턴값이 0일 경우 호출 성공을 의미합니다.

함수 프로토타입

```
ITMGContext ITMGAudioCtrl int AddAudioBlackList(const char* openId)
```

파라미터	유형	의미
openId	char*	블랙리스트에 추가할 ID

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->AddAudioBlackList(openId);
```

가청주파수 데이터 블랙리스트 제거

임의 ID를 가청주파수 데이터 블랙리스트에서 제거합니다. 리턴값이 0일 경우 호출 성공을 의미합니다.

함수 프로토타입

```
ITMGContext ITMGAudioCtrl int RemoveAudioBlackList(const char* openId)
```

파라미터	유형	의미
openId	char*	블랙리스트에서 제거할 ID

예시 코드

```
ITMGContextGetInstance()->GetAudioCtrl()->RemoveAudioBlackList(openId);
```

메시지 콜백

메시지 리스트 :

메시지	메세지 의미
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	오디오 방 메시지 들어가기
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	오디오 방 메시지 나가기
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	방이 네트워크 등 원인으로 메시지 끊김
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	방 유형 변화 이벤트
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	마이크 디바이스 메시지 새로 추가
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	마이크 디바이스 메시지 손실
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	스피커 디바이스 메시지 새로 추가
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	스피커 디바이스 메시지 손실
ITMG_MAIN_EVENT_TYPE_ACCOMPANY_FINISH	반주 종료 메시지
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	방 멤버 업데이트 메시지
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	PTT 녹음 완료
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	PTT 업로드 완료

ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	PTT 다운로드 완료
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	PTT 재생 완료
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	음성 문자 변환 완료

Data 리스트 :

메시지	Data	예제
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_inf
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_inf
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_inf
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; sub_event_type; new_room_type	{"error_inf
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	result; error_info	{"deviceID dd00542b Audio)","e
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	result; error_info	{"deviceID dd00542b Audio)","e
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	result; error_info	{"deviceID 7e454093 Audio)","e
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	result; error_info	{"deviceID 7e454093 Audio)","e
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	user_list; event_id	{"event_ic
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	result; file_path	{"file_path
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	result; file_path;file_id	{"file_id":"
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	result; file_path;file_id	{"file_id":"

ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	result; file_path	{"file_path
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	result; text;file_id	{"file_id":"
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	result; file_path; text;file_id	{"file_id":"

Cocos2D SDK

Project Configuration

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to configure a Cocos2d project for the GME APIs for Cocos2d.

SDK Preparation

1. Download the applicable demo and SDK. For more information, see SDK Download Guide.
2. Decompress the obtained SDK resources.
3. The folder contains:

GMESDK: GME SDK framework file.

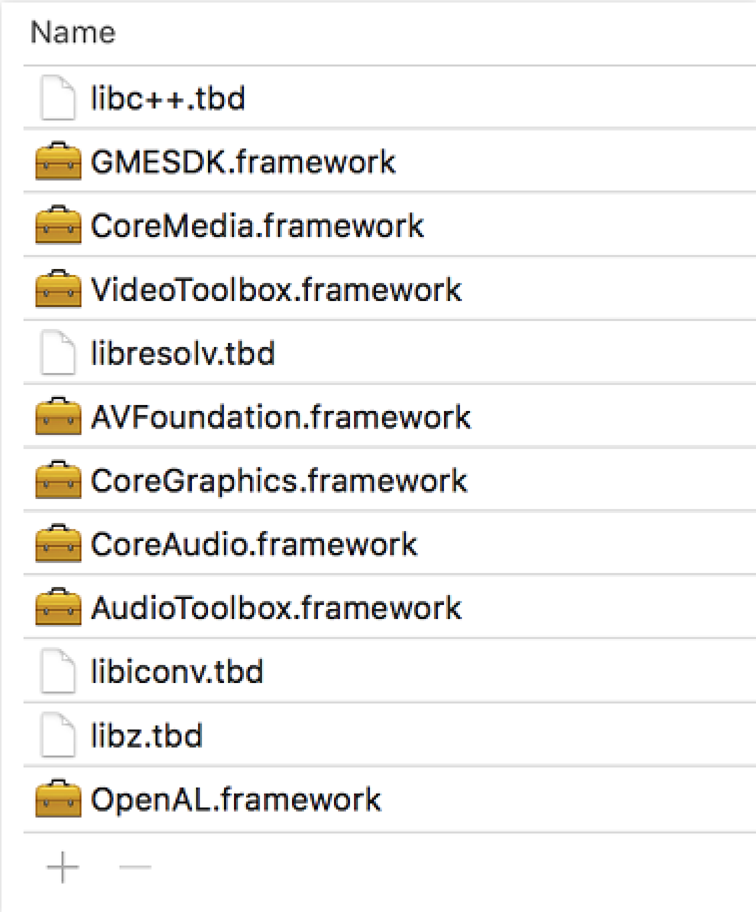
GMECocosDemo: GME SDK demo project.

Note:

The SDK supports compilation on macOS.

iOS Xcode Configuration

1. Add the framework to the Xcode project and set the header file import location (the framework file in the `GMESDK` folder must be added to the project).
2. Add dependent libraries as shown below:



Android Configuration

1. Add `gmesdk.jar` to the libs library.
2. Import the `so` file into `Activity` as shown below:

```
public class AppActivity extends Cocos2dxActivity {
 static final String TAG = "AppActivity";
 static OpensdkGameWrapper gameWrapper ;
 static {
      OpensdkGameWrapper.loadSdkLibrary();
 }
}
```

3. Initialize in the `oncreate` function exactly in the following sequence:

```
protected void onCreate(Bundle savedInstanceState) {
      super.setEnableVirtualButton(false);
      super.onCreate(savedInstanceState);
      // Initialize exactly in the following sequence
      gameWrapper = new OpensdkGameWrapper(this);
```

```
        runOnGLThread(new Runnable() {
            @Override
            public void run() {
                gameWrapper.initOpensdk();
            }
        });
    }
```

4. Configure your project for compilation options by referring to the `Android.mk` in the GME Demo for Cocos.
Path: GMECocos/GMECocosDemo/proj.android-studio/app/jni/Android.mk
Path to the `preBuild.mk` file: /Users/username/Downloads/GMECocos/GMESDK/android/bin/preBuild.mk

Exporting to Different Platforms

Project configuration is required before you can export executables from the Cocos2d engine for different platforms:

Android

Configuring required permissions

Add the following permissions in the `AndroidManifest.xml` file of the project:

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

To use the voice messaging and speech-to-text feature, add the following under the `application` node in the manifest file:

```
<application android:usesCleartextTraffic="true" >
```

Adding permissions as needed

Add the following permissions in the `AndroidManifest.xml` file of the project as needed:
Read/Write permission
Bluetooth permission
The read/write permission is not required. Determine whether to add it according to the following rules:
If you use the default log path (/SDCARD/Android/Data/files), it means that you do not call SetLogPath, and do not need Write_External_Storage permission.
If you call the setLogPath API to set the log path to an external storage device, and the storage path of the voice message recording is an external storage device, you need to apply for the Write_External_Storage permission to the

user and get the user's approval.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Add the Bluetooth permission according to the following rules:

If `targetSDKVersion` in the project is v30 or earlier:

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

If `targetSDKVersion` in the project is v31 or later:

```
<uses-permission android:name="android.permission.BLUETOOTH" android:maxSdkVersion=
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

iOS

Add permissions:

Microphone Usage Description: Allows access to microphone.

Grant the `Allow Arbitrary Loads` permission as shown below:

Key	Type	Value
▼ Information Property List	Dictionary	(19 items)
Localization native development re...	String	$(DEVELOPMENT_LANGUAGE)
Executable file	String	$(EXECUTABLE_NAME)
Bundle identifier	String	$(PRODUCT_BUNDLE_IDENTIFIER)
InfoDictionary version	String	6.0
Bundle name	String	$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
▼ App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES
Privacy - Camera Usage Description	String	Privacy - Camera Usage Description
Privacy - Microphone Usage Desc...	String	NSMicrophoneUsageDescription
▶ Required background modes	Array	(1 item)
Application supports iTunes file sh...	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
Main storyboard file base name	String	Main
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
▶ Supported interface orientations (i...	Array	(4 items)

Windows

You need to download the SDK for Windows as instructed in SDK Download Guide and import it into the project.

Getting Started

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to get started with the GME APIs for Cocos2d.

This document only describes the most important APIs to help you get started with GME. For more information on APIs, please see the API documentation.

Important API	Description
Init	Initializes GME
Poll	Triggers event callback
EnterRoom	Enters room
EnableMic	Enables mic
EnableSpeaker	Enables speaker

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `AV_OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

The `Poll` API should be called periodically for GME to trigger event callbacks.

For GME callback messages, please see the callback message list.

Operation on devices should be performed after successful room entry.

For detailed error codes, please see Error Codes.

Directions for Quick Access

1. Get a singleton

To use the voice feature, get the `ITMGContext` object first.

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->SetTMGDelegate(this);
```

2. Initialize the SDK

For more information on how to get parameters, please see Access Guide.
This API requires the `AppID` from the Tencent Cloud Console and the `openID` as parameters. The `openID` uniquely identifies a user with the rules stipulated by the application developer and must be unique in the application (currently, only INT64 is supported).

Note:

The SDK must be initialized so that a room can be entered

Function prototype

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openID)
```

Parameter	Type	Description
sdkAppId	char*	`AppId` from the Tencent Cloud Console
openId	char*	`OpenID` can only be in Int64 type (converted to char*) with a value greater than 10,000, which is used to identify the user

Sample code

```
#define SDKAPPID3RD "1400089356"
cosnt char* openId="10001";
ITMGContext* context = ITMGContextGetInstance();
context->Init(SDKAPPID3RD, openId);
```

3. Trigger event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update` .

Function prototype

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```

Sample code

```
ITMGContextGetInstance()->Poll();
```

4. Authenticate

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.To get authentication for voice messaging and speech-to-text, the room ID parameter must be set to `null` .

Function prototype

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

Parameter	Type	Description
dwSdkAppID	int	`AppId` from the Tencent Cloud Console.
strRoomID	char*	Room ID, which can contain up to 127 characters (for voice messaging and speech-to-text feature, enter `null`).
strOpenID	char*	User ID, which is the same as `openID` during initialization.
strKey	char*	Permission key from the Tencent Cloud Console.
strAuthBuffer	char*	Returned `authbuff` .
bufferLength	int	Length of the `authbuff` passed in. 500 is recommended.

Sample code

```
unsigned int bufferLen = 512;
unsigned char retAuthBuff[512] = {0};
QAVSDK_AuthBuffer_GenAuthBuffer(atoi(SDKAPPID3RD), roomId, "10001", AUTHKEY,retAuth
```

5. Enter a room

When a client enters a room with the generated authentication information, the `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` message will be received as a callback. Mic and speaker are not enabled by default after room entry. The returned value of `AV_OK` indicates a success.

For entering a common voice chat room that does not involve range voice, use the common room entry API. For more information, please see Range Voice.

Function prototype

```
ITMGContext virtual int EnterRoom(const char*  roomID, ITMG_ROOM_TYPE roomType, con
```

Parameter	Type	Description
roomID	char*	Room ID, which can contain up to 127 characters
roomType	ITMG_ROOM_TYPE	Room audio type
authBuffer	char*	Authentication key
buffLen	int	Authentication key length

For more information on how to choose a room audio type, please see Sound Quality Selection.

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->EnterRoom(roomID, ITMG_ROOM_TYPE_STANDARD, (char*)retAuthBuff,bufferLen);
```

6. Receive callback for room entry

After the client enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        // Process
        break;
        }
    }
}
```

7. Enable/Disable the mic

This API is used to enable/disable the mic. Mic and speaker are not enabled by default after room entry.

Function prototype

```
ITMGAudioCtrl virtual int EnableMic(bool bEnabled)
```

Parameter	Type	Description

bEnabled	bool	To enable the mic, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableMic(true);
```

8. Enable/Disable the speaker

This API is used to enable/disable the speaker.

Function prototype

```
ITMGAudioCtrl virtual int EnableSpeaker(bool enable )
```

Parameter	Type	Description
enable	bool	To disable the speaker, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableSpeaker(true);
```

Voice Chat

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to access and debug GME client APIs for the voice chat feature for Cocos2d.

Key Considerations for Using GME

GME provides the real-time voice, voice message, and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

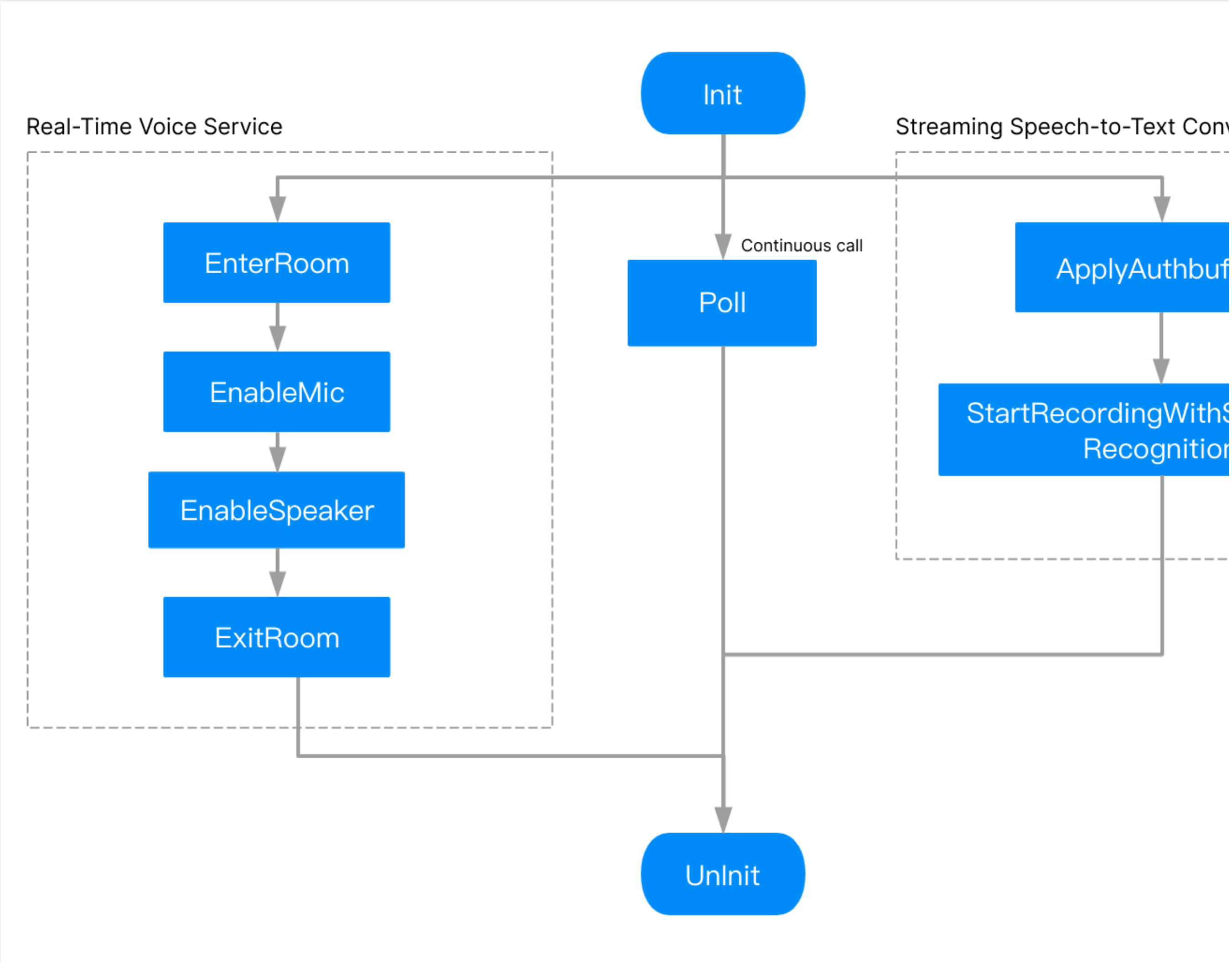
Key notes

You have created a GME application and obtained the `AppID` and `Key` of the SDK as instructed in Activating Services.
You have **activated the real-time voice, voice message, and speech-to-text services of GME** as instructed in Activating Services.
Configure your project before using GME; otherwise, the SDK will not take effect.
After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.
GME APIs should be called in the same thread.
The `Poll` API should be called periodically for GME to trigger event callbacks.
For detailed error code, please see Error Codes.

Connecting to the SDK

Directions

Key processes involved in SDK connection are as follows:



1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Entering a voice chat room, API: EnterRoom
4. Enabling the microphone, API: EnableMic
5. Enabling the speaker, API: EnableSpeaker
6. Exiting a voice room, API: ExitRoom
7. Uninitializing GME, API: UnInit

C++ classes

Class	Description
ITMGContext	Key APIs
ITMGRoom	Room APIs
ITMGRoomManager	Room management APIs

ITMGAudioCtrl	Audio APIs
ITMGAudioEffectCtrl	Sound effect and accompaniment APIs

Key APIs

API	Description
Init	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME

Preparations

You need to import the header file `tmg_sdk.h` first before you can access GME. The classes in the header file inherit `ITMGDelegate` for message delivery and callback.

Sample code

```
#include "tmg_sdk.h"

class TMGTestScene : public cocos2d::Scene,public ITMGDelegate
{
public:
...
private:
...
}
```

Setting a singleton

You need to get `ITMGContext` first before you can call the `EnterRoom` function. All calls begin with `ITMGContext` , which is returned to the application through the `ITMGDelegate` callback and must be set first.

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
```



```
context->SetTMGDelegate(this);
```

Message delivery

The API class uses the `Delegate` method to send callback notifications to the application. `ITMG_MAIN_EVENT_TYPE` indicates the message type. The data on Windows is in json string format. For the key-value pairs, please see the relevant documentation.

Sample code

```
// Function implementation:
//TMGTestScene.h:
class TMGTestScene : public cocos2d::Scene,public ITMGDelegate
{
public:
    void OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data);
}

//TMGTestScene.cpp:
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    // Identify and manipulate `eventType` here
}
```

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

Parameter	Type	Description
sdkAppId	const char*	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	const char*	`openID` can only be in `Int64` type, which is passed in after being converted to a `const char*`. You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
AV_OK = 0	Initialized SDK successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

Notes on 7015 error code:

The 7015 error code is judged by md5. If this error is reported during integration, please check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party reinforcement, Unity packaging mechanism and other factors, the md5 of the library file will be affected, resulting in misjudgment. **Please ignore this error in the logic for official release**, and try to avoid displaying it in the UI.

Sample code

```
#define SDKAPPID3RD "14000xxxxx"
cosnt char* openId="10001";
ITMGContext* context = ITMGContextGetInstance();
context->Init(SDKAPPID3RD, openId);
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update` . `Poll` is the message pump of GME, and the `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.

You can refer to the `EnginePollHelper.cpp` file in the demo.

Calling the `Poll` API periodically:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```


Sample code

```
// Declaration in the header file
class TMGTestScene : public cocos2d::Scene,public ITMGDelegate
{
void update(float delta);
}

// Code implementation
void TMGTestScene::update(float delta)
{
    ITMGContextGetInstance()->Poll();
}
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
ITMGContext int Pause()
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
ITMGContext int Resume()
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

API prototype

```
ITMGContext int Uninit()
```

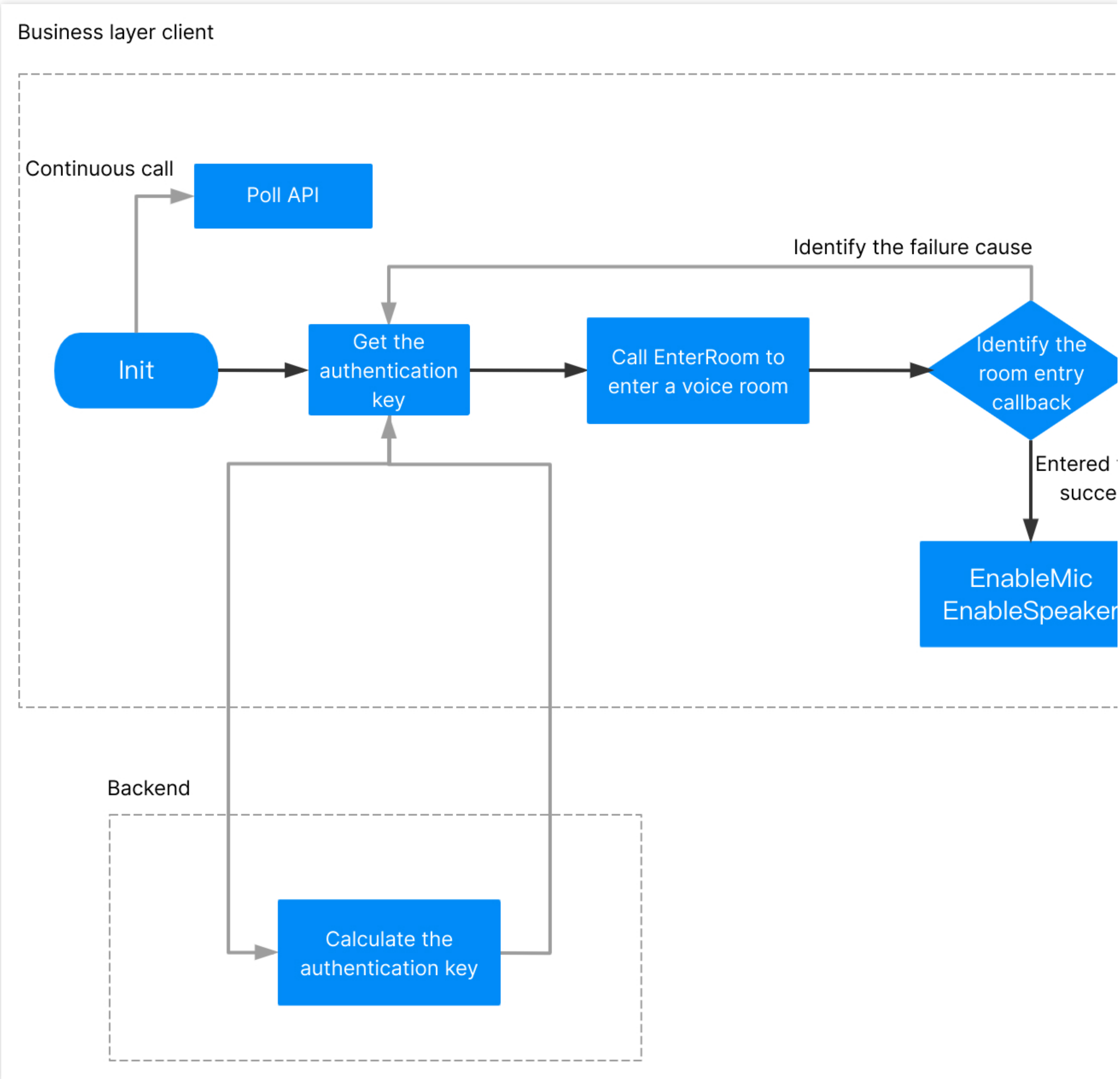
Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->Uninit();
```

Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, please see FAQs About Voice Chat.



API	Description
GenAuthBuffer	Calculates the local authentication key
EnterRoom	Enters a room
ExitRoom	Exits the room
IsRoomEntered	Determines whether room entry is successful
SwitchRoom	Switches the room quickly

Local authentication key calculation

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.

API prototype

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

Parameter	Type	Description
dwSdkAppID	unsigned int	`AppId` from the Tencent Cloud console
strRoomID	const char*	Room ID, which can contain up to 127 characters.
strOpenID	const char*	User ID, which is the same as `openID` during initialization.
strKey	const char*	Permission key from the Tencent Cloud console
strAuthBuffer	const char*	Returned `authbuff`
bufferLength	int	The length of the returned `authbuff` . `500` is recommended.

Sample code

```
unsigned int bufferLen = 512;
unsigned char retAuthBuff[512] = {0};
QAVSDK_AuthBuffer_GenAuthBuffer(atoi(SDKAPPID3RD), roomId, "10001", AUTHKEY,retAuth
```

Entering a room

This API is used to enter a room with the generated authentication information. The mic and speaker are not enabled by default after room entry.

Note：

If the room entry callback result is `0`, the room entry is successful. If `0` is returned from the `EnterRoom` API, it doesn't necessarily mean that the room entry is successful.

The audio type of the room is determined by the first user entering the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will change to the smooth sound quality. Only after a member in the room calls the `ChangeRoomType` API will the audio type of the room be changed.

```
ITMGContext virtual int EnterRoom(const char*  roomID, ITMG_ROOM_TYPE roomType, con
```

Parameter	Type	Description
roomID	const char*	Room ID, which can contain up to 127 characters.
roomType	ITMG_ROOM_TYPE	Room type. We recommend you select `ITMG_ROOM_TYPE_FLUENCY` for games. For more information on room audio types, see Sound Quality.
authBuffer	const char*	Authentication key
buffLen	int	Authentication key length

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->EnterRoom(roomID, ITMG_ROOM_TYPE_FLUENCY, (char*)retAuthBuff,bufferLen);
```

Callback for room entry

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function for callback and processing. A successful callback means that the room entry is successful, and the billing starts.

Billing references

Purchase Guide

Billing FAQs

Will Voice Chat still be charged when client is offlined?

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        // Process
        break;
        }
    }
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnected callback prompt `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` . At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` . When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` .

Error codes

Error Code Value	Cause and Suggested Solution
7006	Authentication failed: The `AppID` does not exist or is incorrect. An error occurred while authenticating the `authbuff` . Authentication expired. The `OpenId` does not meet the specification.
7007	Already in another room.
1001	The user was already in the process of entering a room but repeated this operation. It is recommended not to call the room entering API until the room entry callback is returned.
1003	The user was already in the room and called the room entering API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally

Exiting a room

This API is used to exit the current room. It is an async API. The returned value `AV_OK` indicates a successful async delivery. If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API during API call; instead, you can directly call the `EnterRoom` API.

API prototype

```
ITMGContext virtual int ExitRoom()
```

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->ExitRoom();
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM`.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
        // Process
        break;
        }
    }
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}

Determining whether a user has entered a room

This API is used to determine whether the user has entered a room. A value in bool type will be returned. Do not call this API during room entry.

API prototype

```
ITMGContext virtual bool IsRoomEntered()
```

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->IsRoomEntered();
```

Switching room

User can call this API to quickly switch the voice chat room after entering the room. After the room is switched, the device is not reset, that is, if the microphone is already enabled in this room, the microphone will keep enabled after the room is switched.

The callback for quickly switching rooms is `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM` , and the fields are `error_info` and `result` .

API prototype

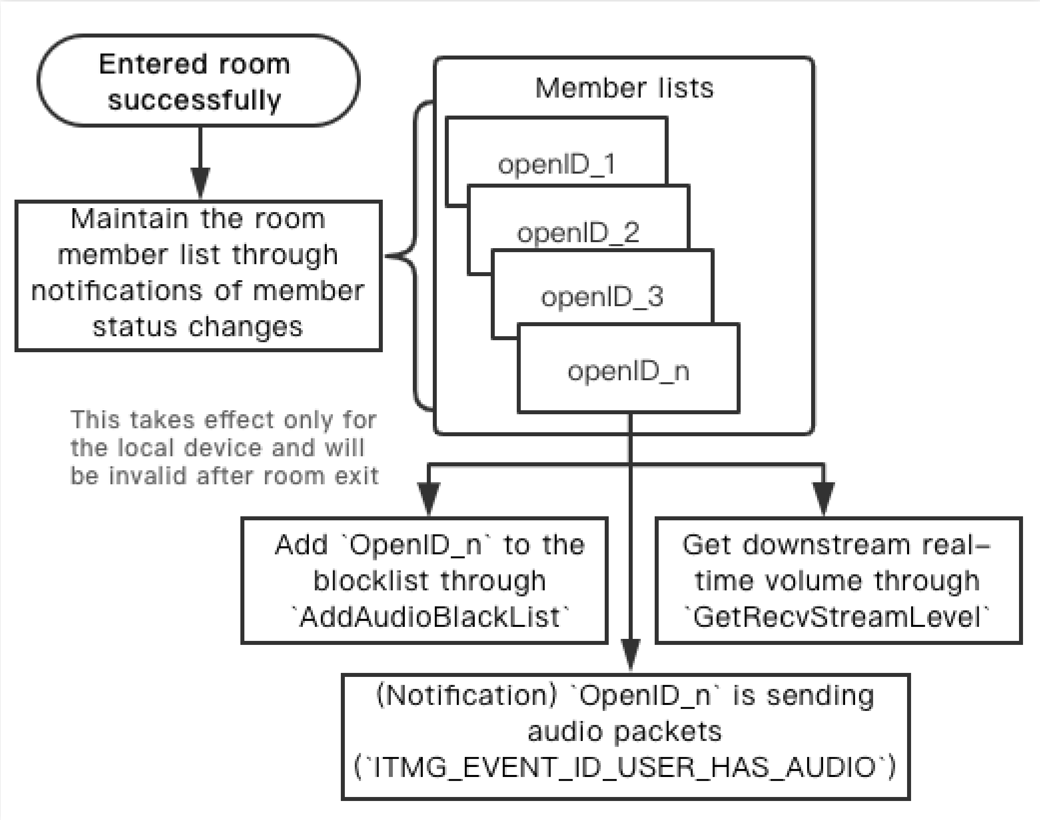
```
ITMGContext virtual int SwitchRoom(const char* targetRoomID, const char* authBuff,
```

Type descriptions

Parameter	Type	Description
targetRoomID	const char*	ID of the room to enter
authBuffer	const char*	Generates a new authentication key with the ID of the room to enter
buffLen	int	Authentication key length

Room Status Maintenance

APIs in this section are used to display speaking members and members entering or exiting the room and mute a member in the room at the business layer.



API/Notification	Description
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The member status changed
AddAudioBlackList	Mutes a member in the room
RemoveAudioBlackList	Unmutes a user

Notification events of member room entry and speaking status

This API is used to obtain the user speaking in the room and display it in the UI, and to send a notification when someone entering or exiting the room.

Notification for this event will be sent only when the status changes. To get member status in real time, cache the notification when it is received at a higher layer. The event message is `ITMG_MAIN_EVNET_TYPE_USER_UPDATE` , where the data contains `event_id` and `user_list` . The event message will be identified in the `OnEvent` function.

Notifications for audio events are subject to a threshold, and a notification will be sent only when this threshold is exceeded. The notification "A member has stopped sending audio packets" will be sent if no audio packets are received in more than two seconds.

event_id	Description	Maintenance
ITMG_EVENT_ID_USER_ENTER	Return the `openid` of the member entering the	Member list

	room.	
ITMG_EVENT_ID_USER_EXIT	Return the `openid` of the member exiting the room.	Member list
ITMG_EVENT_ID_USER_HAS_AUDIO	Return the `openid` of the member sending audio packets in the room. This event can be used to determine whether a user is speaking and display the voiceprint effect.	Chat member list
ITMG_EVENT_ID_USER_NO_AUDIO	Return the `openid` of the member stopping sending audio packets in the room.	Chat member list

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVNET_TYPE_USER_UPDATE:
        {
        // Process
        // Parse the parameter to get `eventID` and `user_list`
            switch (eventID)
             {
             case ITMG_EVENT_ID_USER_ENTER:
                  // A member enters the room
                  break;
             case ITMG_EVENT_ID_USER_EXIT:
                  // A member exits the room
                break;
            case ITMG_EVENT_ID_USER_HAS_AUDIO:
                // A member sends audio packets
                break;
            case ITMG_EVENT_ID_USER_NO_AUDIO:
                // A member stops sending audio packets
                break;
             default:
                break;
            }
        break;
        }
    }
}
```

Muting a member in the room

This API is used to add an ID to the audio data blocklist. This operation blocks audio from someone and only applies to the local device without affecting other devices. The returned value `0` indicates that the call is successful.

Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B;

If B blocks neither A nor C, B can hear both of them;

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

API prototype

```
ITMGContext ITMGAudioCtrl int AddAudioBlackList(const char* openId)
```

Parameter	Type	Description
openId	char*	`openid` of the user to be blocked

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->AddAudioBlackList(openId);
```

Unmuting

This API is used to remove an ID from the audio data blacklist. A returned value of 0 indicates the call is successful.

API prototype

```
ITMGContext ITMGAudioCtrl int RemoveAudioBlackList(const char* openId)
```

Parameter	Type	Description
openId	char*	ID to be unblocked

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->RemoveAudioBlackList(openId);
```

Voice Chat Capturing APIs

The voice chat APIs can only be called after SDK initialization and room entry.

When the user clicks the button of enabling/disabling the mic or speaker on the UI, we recommend you call the `EnableMic` or `EnableSpeaker` API.

To enable the user to press the mic button on the UI to speak and release it to stop speaking, we recommend you call `EnableAudioCaptureDevice` once during room entry and call `EnableAudioSend` to enable the user to speak while pressing the button.

API	Description
EnableMic	Enables/Disables the mic
GetMicState	Gets the mic status
EnableAudioCaptureDevice	Enables/Disables the capturing device
IsAudioCaptureDeviceEnabled	Gets the capturing device status
EnableAudioSend	Enables/Disables audio upstreaming
IsAudioSendEnabled	Gets the audio upstreaming status
GetMicLevel	Gets the real-time mic volume level
GetSendStreamLevel	Gets real-time audio upstreaming volume
SetMicVolume	Sets the mic volume level
GetMicVolume	Gets the mic volume level

Enabling or disabling mic

This API is used to enable/disable the mic. The mic and speaker are not enabled by default after room entry.

EnableMic = EnableAudioCaptureDevice + EnableAudioSend

API prototype

```
ITMGAudioCtrl virtual int EnableMic(bool bEnabled)
```

Parameter	Type	Description
bEnabled	bool	To enable the mic, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable mic
ITMGContextGetInstance()->GetAudioCtrl()->EnableMic(true);
```


Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

API prototype

```
ITMGAudioCtrl virtual int GetMicState()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicState();
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.
This API can only be called after room entry. The device will be disabled automatically after room exit.
Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

API prototype

```
ITMGAudioCtrl virtual int EnableAudioCaptureDevice(bool enable)
```

Parameter	Type	Description
enable	bool	To enable the capturing device, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable capturing device
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioCaptureDevice(true);
```

Getting the capturing device status

This API is used to get the status of a capturing device.

API prototype

```
ITMGContext virtual bool IsAudioCaptureDeviceEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioCaptureDeviceEnabled();
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain mute. For more information on how to enable/disable the capturing device, please see the `EnableAudioCaptureDevice` API.

API prototype

```
ITMGContext  virtual int EnableAudioSend(bool bEnable)
```

Parameter	Type	Description
bEnable	bool	To enable audio upstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioSend(true);
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

API prototype

```
ITMGContext virtual bool IsAudioSendEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioSendEnabled();
```

Getting the real-time mic volume

This API is used to get the real-time mic volume. An int-type value in the range of 0-100 will be returned. It is recommended to call this API once every 20 ms.

API prototype

```
ITMGAudioCtrl virtual int GetMicLevel()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicLevel();
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume. An int-type value in the range of 0-100 will be returned.

API prototype

```
ITMGAudioCtrl virtual int GetSendStreamLevel()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSendStreamLevel();
```

Setting the mic software volume

This API is used to set the mic volume level. The corresponding parameter is `volume` , which is equivalent to attenuating or gaining the captured sound.

API prototype

```
ITMGAudioCtrl virtual int SetMicVolume(int vol)
```

Parameter	Type	Description
vol	int	Value range: 0–200. Default value: 100. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
int micVol = (int)(value * 100);
ITMGContextGetInstance()->GetAudioCtrl()->SetMicVolume(vol);
```

Getting the mic software volume

This API is used to obtain the microphone volume. An "int" value is returned. Value 101 represents API SetMicVolume has not been called.

API prototype

```
ITMGAudioCtrl virtual int GetMicVolume()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicVolume();
```

Voice Chat Playback APIs

API	Description
EnableSpeaker	Enables/Disables the speaker
GetSpeakerState	Gets the speaker status
EnableAudioPlayDevice	Enables/Disables the playback device
IsAudioPlayDeviceEnabled	Gets playback device status
EnableAudioRecv	Enables/Disables audio downstreaming
IsAudioRecvEnabled	Gets the audio downstreaming status
GetSpeakerLevel	Gets the real-time speaker volume level
GetRecvStreamLevel	Gets the real-time downstreaming audio volume levels of other members in the room
SetSpeakerVolume	Sets the speaker volume level
GetSpeakerVolume	Gets the speaker volume level

Enabling or disabling speaker

This API is used to enable/disable the speaker. **EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv**

API prototype

```
ITMGAudioCtrl virtual int EnableSpeaker(bool enable)
```

Parameter	Type	Description
enable	bool	To disable the speaker, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
// Enable the speaker
ITMGContextGetInstance()->GetAudioCtrl()->EnableSpeaker(true);
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, and 1 is on.

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerState()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerState();
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

API prototype

```
ITMGAudioCtrl virtual int EnableAudioPlayDevice(bool enable)
```

Parameter	Type	Description
enable	bool	To disable the playback device, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioPlayDevice(true);
```

Getting the playback device status

This API is used to get the status of a playback device.

API prototype

```
ITMGAudioCtrl virtual bool IsAudioPlayDeviceEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioPlayDeviceEnabled();
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain mute. For more information on how to enable/disable the playback device, please see the `EnableAudioPlayDevice` API.

API prototype

```
ITMGAudioCtrl virtual int EnableAudioRecv(bool enable)
```

Parameter	Type	Description
enable	bool	To enable audio downstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioRecv(true);
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

API prototype

```
ITMGAudioCtrl virtual bool IsAudioRecvEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioRecvEnabled();
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume. An int-type value will be returned to indicate the volume. It is recommended to call this API once every 20 ms.

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerLevel()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerLevel();
```

Getting the real-time downstreaming audio levels of other members in the room

This API is used to get the real-time audio downstreaming volume of other members in the room. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGAudioCtrl virtual int GetRecvStreamLevel(const char* openId)
```

Parameter	Type	Description
openId	char*	`openId` of other members in the room

Sample code

```
iter->second.level = ITMGContextGetInstance()->GetAudioCtrl()->GetRecvStreamLevel(i
```

Setting the speaker volume

This API is used to set the speaker volume.

API prototype

```
ITMGAudioCtrl virtual int SetSpeakerVolume(int vol)
```

Parameter	Type	Description
vol	int	Volume level. Value range: 0–200. Default value: 100. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
int vol = 100;
ITMGContextGetInstance()->GetAudioCtrl()->SetSpeakerVolume(vol);
```

Getting the speaker volume

This API is used to get the speaker volume. An int-type value will be returned to indicate the volume. 101 indicates that the `SetSpeakerVolume` API has not been called.

"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerVolume()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerVolume();
```

Device Selection APIs

Device selection APIs can be used only on PC.

API	Description
GetMicListCount	Gets the number of mics
GetMicList	Lists mics
GetSpeakerListCount	Gets the number of speakers
GetSpeakerList	Lists speakers
SelectMic	Selects mics
SelectSpeaker	Selects speakers

Getting the number of mics

This API is used to get the number of mics.

Function prototype

```
ITMGAudioCtrl virtual int GetMicListCount()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicListCount();
```

Enumerating mics

This API is used together with the `GetMicListCount` API to enumerate mics.

Function prototype

```
ITMGAudioCtrl virtual int GetMicList(TMGAudioDeviceInfo* ppDeviceInfoList, int nCou

class TMGAudioDeviceInfo
{
public:
    const char* pDeviceID;
    const char* pDeviceName;
};
```

Parameter	Type	Description
ppDeviceInfoList	TMGAudioDeviceInfo	Device list
nCount	int	Number of the mics

`TMGAudioDeviceInfo` Parameter	Type	Description
pDeviceID	const char*	Device ID
pDeviceName	const char*	Device name

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicList(ppDeviceInfoList,nCount);
```

Selecting a mic

This API is used to select a mic. If this API is not called or `DEVICEID_DEFAULT` is passed in, the default mic will be selected.

The 0th device id returned in the GetMicList API is the default device of the call device. If there is a selected call device, it will be maintained by service. If it is unplugged, the call device will be changed back into the default device.

Function prototype

```
ITMGAudioCtrl virtual int SelectMic(const char* pMicID)
```

Parameter	Type	Description
pMicID	const char*	Mic ID, which is from the list returned by `GetMicList` .

Sample code

```
const char* pMicID ="{0.0.1.00000000}.{7b0b712d-3b46-4f7a-bb83-bf9be4047f0d}";
ITMGContextGetInstance()->GetAudioCtrl()->SelectMic(pMicID);
```

Getting the number of speakers

This API is used to get the number of speakers.

Function prototype

```
ITMGAudioCtrl virtual int GetSpeakerListCount()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerListCount();
```

Enumerating speakers

This API is used together with the `GetSpeakerListCount` API to enumerate speakers.

Function prototype

```
ITMGAudioCtrl virtual int GetSpeakerList(TMGAudioDeviceInfo* ppDeviceInfoList, int
```

Parameter	Type	Description
ppDeviceInfoList	TMGAudioDeviceInfo	Device list
nCount	int	Number of the speakers

`TMGAudioDeviceInfo` Parameter	Type	Description
pDeviceID	const char*	Device ID

pDeviceName	const char*	Device name

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerList(ppDeviceInfoList,nCount);
```

Selecting a speaker

This API is used to select a playback device. If this API is not called or `DEVICEID_DEFAULT` is passed in, the default playback device will be selected.

Function prototype

```
ITMGAudioCtrl virtual int SelectSpeaker(const char* pSpeakerID)
```

Parameter	Type	Description
pSpeakerID	const char*	Speaker ID, which is from the list returned by `GetSpeakerList` .

Sample code

```
const char* pSpeakerID ="{0.0.1.00000000}.{7b0b712d-3b46-4f7a-bb83-bf9be4047f0d}";
ITMGContextGetInstance()->GetAudioCtrl()->SelectSpeaker(pSpeakerID);
```

Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

API prototype

```
ITMGAudioCtrl virtual int EnableLoopBack(bool enable)
```

Parameter	Type	Description
enable	bool	Specifies whether to enable

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableLoopBack(true);
```

Getting a user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, please see the `EnterRoom` API.

API prototype

```
class ITMGRoom {
public:
    virtual ~ITMGRoom() {} ;
    virtual int GetRoomType() = 0;

};
```

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
ITMGContextGetInstance()->GetRoom()->GetRoomType();
```

Changing the room type

This API is used to modify a user's room audio type. For the result, please see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

API prototype

```
IITMGContext TMGRoom public int ChangeRoomType((ITMG_ROOM_TYPE roomType)
```

Parameter	Type	Description
roomType	ITMG_ROOM_TYPE	Room type to be switched to the target type. For room audio types, please see the `EnterRoom` API.

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
```



```
ITMGContextGetInstance()->GetRoom()->ChangeRoomType(ITMG_ROOM_TYPE_FLUENCY);
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; new_room_type	{"error_info":"","new_room_type":0

Callback for room type setting completion

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result` , `error_info` , and `new_room_type` . The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	Indicates that the existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	Indicates that a user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	Indicates that a user is already in the room and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	Indicates that a room member calls the `ChangeRoomType` API to request a change of room audio type.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE == t
        {
        // Process room type events
      }
}
```

The monitoring event of room call quality

This is the quality monitoring event, which is triggered once every two seconds after room entry, and its message is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY`. The returned parameters include `weight`, `loss`, and `delay`, which are as detailed below:

Parameter	Type	Description
weight	int	Value range: 1-50. 50 indicates excellent sound quality, 1 indicates very poor (barely usable) sound quality, and 0 represents an initial meaningless value. Generally, if the value is below 30, the business layer will remind users that the network is poor and recommend them to switch the network.
loss	double	Upstream packet loss rate
delay	int	Voice chat delay in ms

Getting the version number

This API is used to get the SDK version number for analysis.

API prototype

```
ITMGContext virtual const char* GetSDKVersion()
```

Sample code

```
ITMGContextGetInstance()->GetSDKVersion();
```

Setting the log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
ITMGContext int SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO

levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

`ITMG_LOG_LEVEL` is as detailed below:

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContextGetInstance()->SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

OS	Path
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API prototype

```
ITMGContext virtual int SetLogPath(const char* logDir)
```

Parameter	Type	Description
logDir	const char*	Path

Sample code

```
cosnt char* logDir = ""// Set a path by yourself

ITMGContext* context = ITMGContextGetInstance();
context->SetLogPath(logDir);
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

API prototype

```
ITMGRoom virtual const char* GetQualityTips()
```

Sample code

```
ITMGContextGetInstance()->GetRoom()->GetQualityTips();
```

Callback Messages

Message	Description	Parameter
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected for network or other reasons	result; error
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The room members	user_list; event_id

	were updated	
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	Room reconnection started	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	Room reconnection succeeded	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status changed	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker was changed	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A speaker was added	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A mic was added	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic was lost	result; error

ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic was changed	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	Room quality message	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Recording of a voice message was completed	result; file_
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Upload of a voice message was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Download of a voice message was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Playback of a voice message was completed	result; file_
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast recording-to-text conversion was completed	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed	result; file_ text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	A voice message is	result; file_ text;file_id

	being converted into text in a streaming manner	
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed	result; text;file_id

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to integrate with and debug GME client APIs for the voice messaging and speech-to-text services for Cocos2d.

Key Considerations for Using GME

GME provides the real-time voice service and voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

Caution:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.

Non-streaming speech-to-text API ***SpeechToText()***: There can be up to 10 concurrent requests per account.

Streaming speech-to-text API ***StartRecordingWithStreamingRecognition()***: There can be up to 50 concurrent requests per account.

Real-time streaming speech-to-text API ***StartRealTimeASR()***: There can be up to 50 concurrent requests per account.

Notes

You have created a GME application and obtained the SDK AppID and key. For more information, see Activating Services.

You have activated **GME real-time voice service and voice messaging and speech-to-text services**. For more information, see Activating Services.

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `AV_OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

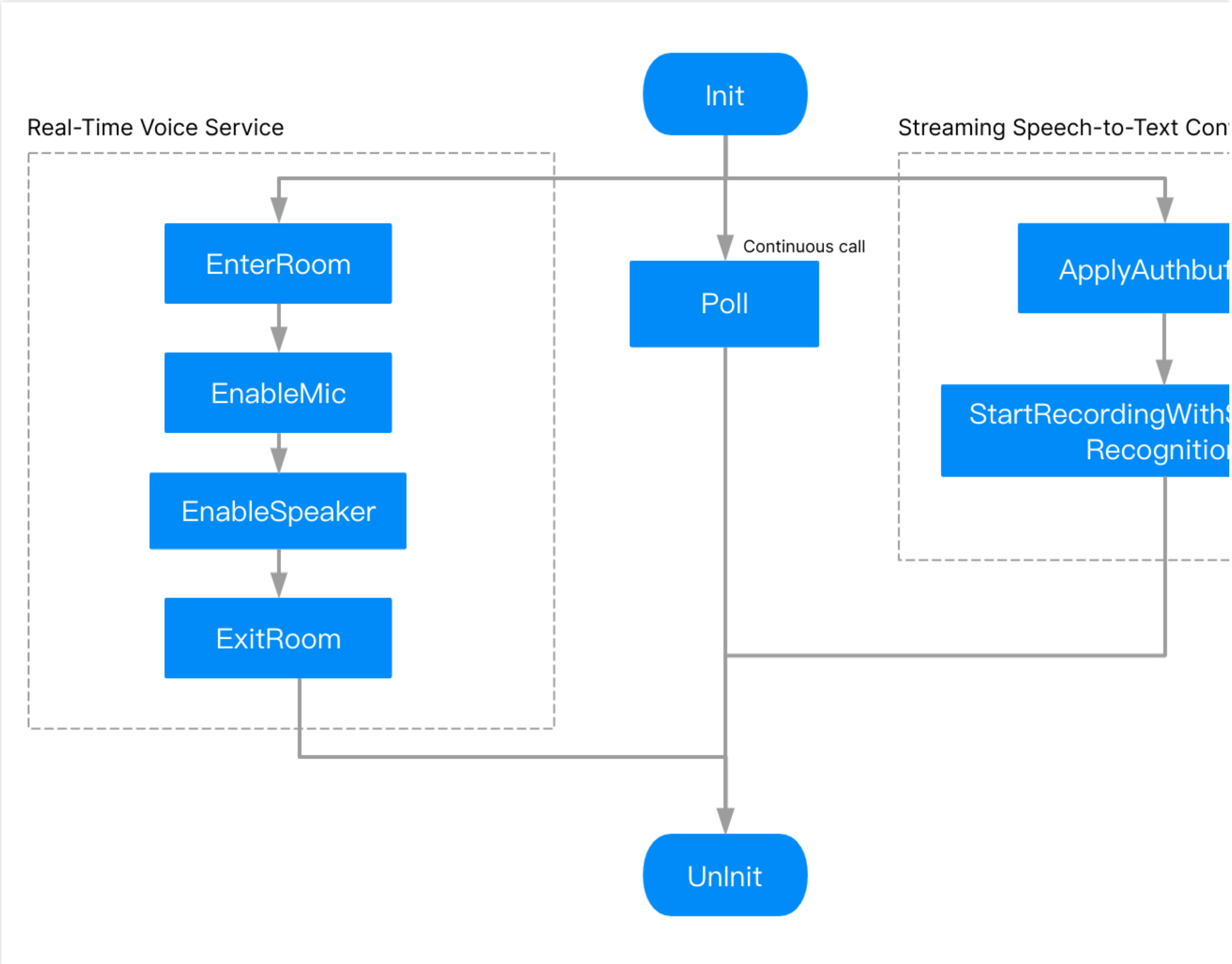
The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error code, see Error Codes.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Initializing authentication, API: ApplyPTTAuthbuffer
4. Starting streaming speech recognition, API: StartRecordingWithStreamingRecognition
5. Stop recording, API: StopRecording
6. Uninitializing GME, API: UnInit

C++ classes

Class	Description
ITMGContext	Key APIs
ITMGPTT	Voice messaging and speech-to-text APIs

Key APIs

API	Description
Init	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME

Importing the header file

```
#include "auth_buffer.h"
#include "tmg_sdk.h"
#include "AdvanceHeaders/tmg_sdk_adv.h"
#include <vector>
```

Callback

Setting callback sample

```
// When initializing the SDK
m_pTmgContext = ITMGContextGetInstance();
m_pTmgContext->SetTMGDelegate(this);

// In the destructor
CTMGSDK_For_AudioDlg::~CTMGSDK_For_AudioDlg()
{
            ITMGContextGetInstance()->SetTMGDelegate(NULL);
}
```

Message delivery

The API class uses the `Delegate` method to send callback notifications to the application. `ITMG_MAIN_EVENT_TYPE` indicates the message type. The data on Windows is in json string format. For the key-value pairs, please see the relevant documentation.

```
// Declaration in the header file
virtual void OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data);
```



```
// Sample code
void CTMGSDK_For_AudioDlg::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data
{
            switch(eventType)
            {
            case ITMG_MAIN_EVENT_TYPE_XXXX_XXXX:
                {
                    // Process the callback
                }
                break;
            }
}
```

Getting singleton

The GME SDK is provided in the form of a singleton, all calls begin with `ITMGContext` , and callbacks are passed to the application through `ITMGDelegate` , which should be configured first.

Sample code

```
ITMGContext* m_pTmgContext;
m_pTmgContext->Init(AppID, OpenID);
```

Initializing SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

Parameter	Type	Description
sdkAppId	const char*	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	const char*	`openID` can only be in `Int64` type, which is passed in after being converted to a `const char*` . You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
AV_OK = 0	Initialized SDK successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

Notes on 7015 error code:

The 7015 error code is judged by md5. If this error is reported during integration, please check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party reinforcement, Unity packaging mechanism and other factors, the md5 of the library file will be affected, resulting in misjudgment. **Please ignore this error in the logic for official release**, and try to avoid displaying it in the UI.

Sample code

```
#define SDKAPPID3RD "14000xxxxxx"
cosnt char* openId="10001";
ITMGContext* context = ITMGContextGetInstance();
context->Init(SDKAPPID3RD, openId);
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update` . `Poll` is the message pump of GME, and the `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.

You can refer to the `EnginePollHelper.cpp` file in the demo.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```

Sample code

```
void TMGTestScene::update(float delta)
{
    ITMGContextGetInstance()->Poll();
}
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
ITMGContext int Pause()
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
ITMGContext int Resume()
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid` , switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

API prototype

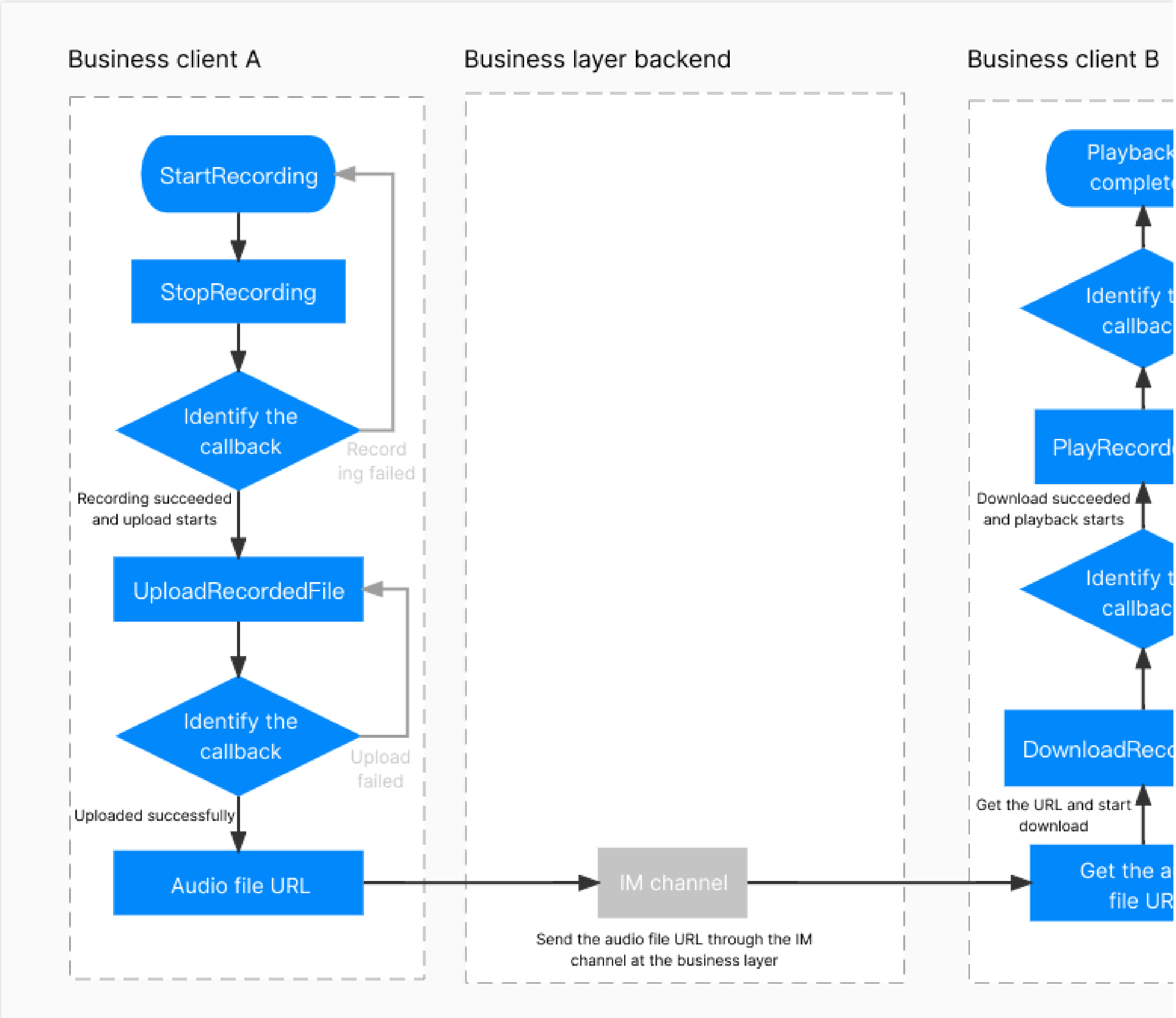
```
ITMGContext int Uninit()
```

Voice Messaging and Speech-to-Text Services

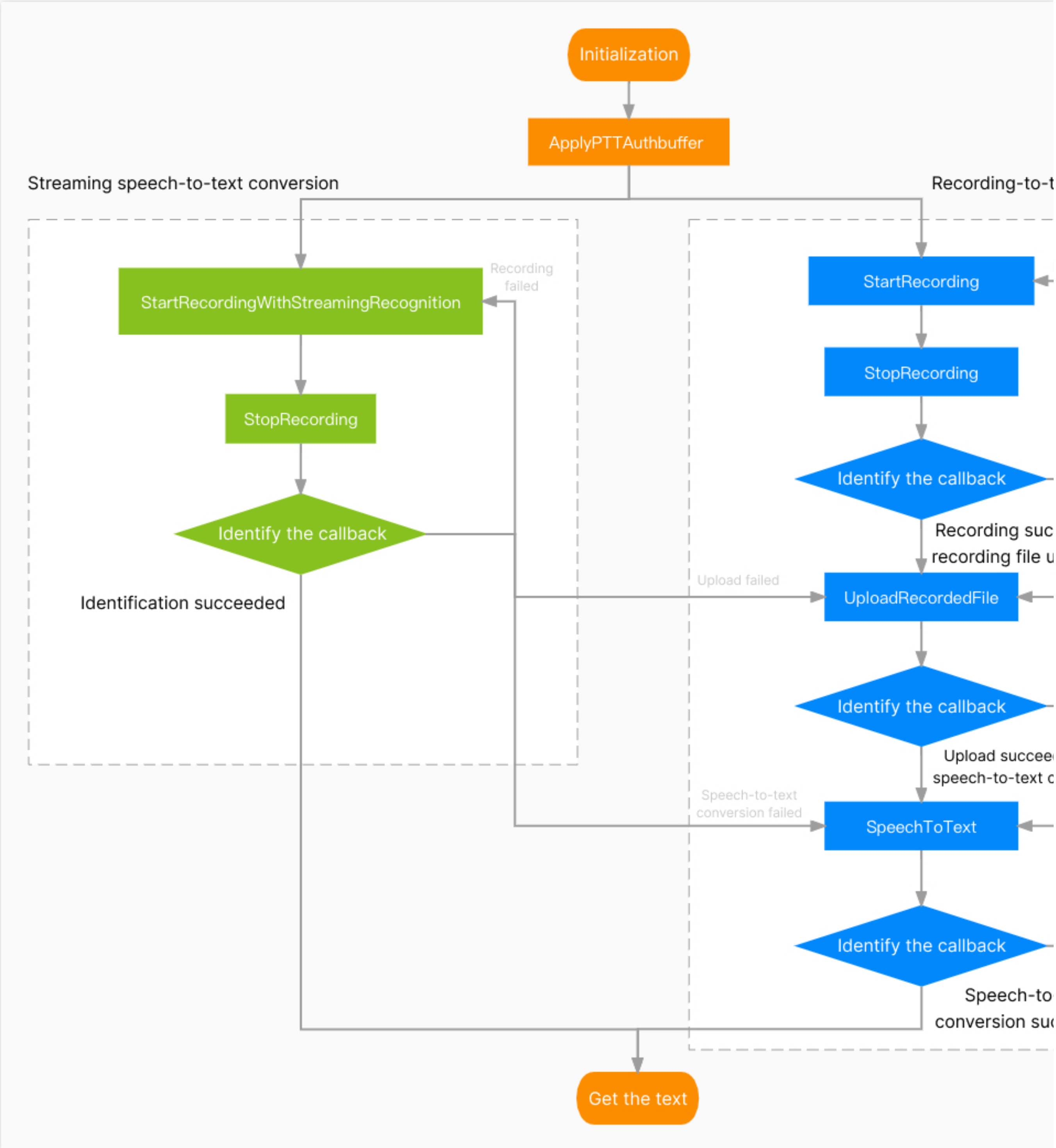
Note:

The speech-to-text service consists of fast recording-to-text conversion and streaming speech-to-text conversion. You do not need to enter a voice chat room when using the voice message service.

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, call the `SetMaxMessageLength` API to set it after initialization.



Flowchart for using the speech-to-text service



API	Description
GenAuthBuffer	Generates the local authentication key
ApplyPTTAuthbuffer	Initializes authentication
SetMaxMessageLength	Specifies the maximum length of voice message

Generating the local authentication key

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.

API prototype

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

Parameter	Type	Description
dwSdkAppID	int	`AppId` from the Tencent Cloud console.
strRoomID	const char*	Enter `null` or an empty string
strOpenID	const char*	User Identifier, which is the same as `openID` during initialization.
strKey	const char*	Permission key from the Tencent Cloud console.
strAuthBuffer	const char*	Returned `authbuff` .
bufferLength	int	Length of the `authbuff` passed in. 500 is recommended.

Application authentication

After the authentication information is generated, the authentication is assigned to the SDK.

API prototype

```
ITMGPTT virtual int ApplyPTTAuthbuffer(const char* authBuffer, int authBufferLen)
```

Parameter	Type	Description
authBuffer	const char*	Authentication
authBufferLen	int	Authentication length

Sample code

```
ITMGContextGetInstance()->GetPTT()->ApplyPTTAuthbuffer(authBuffer,authBufferLen);
```

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

API prototype

```
ITMGPTT virtual int SetMaxMessageLength(int msTime)
```

Parameter	Type	Description
msTime	int	Audio duration in ms. Value range: 1000 < msTime <= 58000

Sample code

```
int msTime = 10000;
ITMGContextGetInstance()->GetPTT()->SetMaxMessageLength(msTime);
```

Streaming Speech Recognition

Voice messaging and speech-to-text APIs

API	Description
StartRecordingWithStreamingRecognition	Starts streaming recording
StopRecording	Stops recording

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call** Stop recording.

API prototype

```
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath)
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath,con
```

Parameter	Type	Description
filePath	const char*	Path of stored audio file
speechLanguage	const char*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translateLanguage	const char*	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContextGetInstance()->GetPTT()->StartRecordingWithStreamingRecognition(filePath
```

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen on callback messages in the `OnEvent` notification, which is as detailed below:

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real-time during the recording, which is equivalent to returning the recognized text while speaking.

The event message will be identified in the `OnEvent` notification based on the actual needs. The passed parameters include the following four messages.

Message Name	Description
result	Return code indicating whether streaming speech recognition is successful
text	Text converted from speech
file_path	Local path of stored recording file
file_id	Backend URL address of recording file, which will be retained for 90 days

Note:

The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Suggested Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text	The message returned contains a backend URL after

	conversion failed, but recording and upload succeeded.	successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

If the error code 4098 is reported, see FAQs for solutions.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
                switch (eventType) {
                    case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
                    {
                        HandleSTREAM2TEXTComplete(data,true);
                        break;
                        }
                    ...
                            case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_R
                    {
                        HandleSTREAM2TEXTComplete(data, false);
                        break;
                    }
                }
}

void CTMGSDK_For_AudioDlg::HandleSTREAM2TEXTComplete(const char* data, bool isCompl
{
                    std::string strText = "STREAM2TEXT: ret=";
                    strText += data;
                    m_EditMonitor.SetWindowText(MByteToWChar(strText).c_str());
                    Json::Reader reader;
                    Json::Value root;
                    bool parseRet = reader.parse(data, root);
                    if (!parseRet) {
                        ::SetWindowText(m_EditInfo.GetSafeHwnd(),MByteToWChar(std:::
                    }
```

```
                        else
                        {
                            if (isComplete) {
                                ::SetWindowText(m_EditUpload.GetSafeHwnd(), MByteToWCha
                            }
                            else {
                                std::string isruning = "STREAMINGRECOGNITION_IS_RUNNING
                                ::SetWindowText(m_EditUpload.GetSafeHwnd(), MByteToWCha
                            }
                        }
}
```

Voice Message Recording

The recording process is as follows: start recording > stop recording > return recording callback > start the next recording.

Voice message and speech-to-text APIs

API	Description
StartRecording	Starts recording
PauseRecording	Pauses recording
ResumeRecording	Resumes recording
StopRecording	Stops recording
CancelRecording	Cancels recording

Starting recording

This API is used to start recording.

API prototype

```
ITMGPTT virtual int StartRecording(const char* fileDir)
```

Parameter	Type	Description
fileDir	const char*	Path of stored audio file

Sample code

```
char buffer[256]={0};
snprintf(buffer, sizeof(buffer), "%sunreal_ptt_local.file", getFilePath().c_str());
ITMGContextGetInstance()->GetPTT()->StartRecording(buffer);
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

API prototype

```
ITMGPTT virtual int StopRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->StopRecording();
```

Callback for recording start

The recording start result will be returned through the callback.
To stop recording, call `StopRecording`. The callback for recording start will be returned after the recording is stopped.

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file, which must be accessible and cannot be the `fileid`

Error codes

Error Code Value	Cause	Suggested Solution
4097	Parameter is empty.	Check whether the API parameters in the code are correct.
4098	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
4099	Recording is in progress.	Ensure that the SDK recording feature is used at the right time.

4100	Audio data is not captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	Mic permission is required for using the SDK. To add the permission, please see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
                switch (eventType) {
                    case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
                    {
                    // Process
                    break;
                        }
                    ...
                            case ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
                    {
                    // Process
                    break;
                    }
                }
}
```

Pausing recording

This API is used to pause recording. If you want to resume recording, please call the `ResumeRecording` API.

API prototype

```
ITMGPTT virtual int PauseRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->PauseRecording();
```

Resuming recording

This API is used to resume recording.

API prototype

```
ITMGPTT virtual int ResumeRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->ResumeRecording();
```

Canceling recording

This API is used to cancel recording. **There is no callback after cancellation**.

API prototype

```
ITMGPTT virtual int CancelRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->CancelRecording();
```

Voice Message Upload, Download, and Playback

API	Description
UploadRecordedFile	Uploads the audio file
DownloadRecordedFile	Downloads the audio file
PlayRecordedFile	Plays back the audio file
StopPlayFile	Stops playing back the audio file
GetFileSize	Gets audio file size
GetVoiceFileDuration	Gets the audio file duration

Uploading an audio file

This API is used to upload an audio file.

API prototype

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of uploaded audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->UploadRecordedFile(filePath);
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file
fileid	FString	File URL path

Error codes

Error Code Value	Cause	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.
8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
8195	A network error occurred.	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.

8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
        switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
            ...
                    case ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
            {
            // Process
            break;
            }
        }
}
```

Downloading the audio file

This API is used to download an audio file.

API prototype

```
ITMGPTT virtual int DownloadRecordedFile(const char* fileId, const char* filePath)
```

Parameter	Type	Description
fileId	const char*	URL path of file
filePath	const char*	Local path of saved file

Sample code


```
ITMGContextGetInstance()->GetPTT()->DownloadRecordedFile(fileID,filePath);
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file
fileid	FString	URL path of file, which will be retained on the server for 90 days

Error codes

Error Code Value	Cause	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
12291	Network storage system exception.	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	Server file system error.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed during the process of getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.
12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.

12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
        switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
```

Playing back audio

This API is used to play back audio.

API prototype

```
ITMGPTT virtual int PlayRecordedFile(const char* filePath)
ITMGPTT virtual int PlayRecordedFile(const char* filePath, nt voiceType)
```

Parameter	Type	Description
filePath	const char*	Local audio file path
voicetype	int	Voice changer type. For more information, see Voice Changing Effects.

Error codes

Error Code Value	Cause	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
ITMGContextGetInstance()->GetPTT()->PlayRecordedFile(filePath);
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameter includes `result` and `file_path`.

Parameter	Type	Description
code	int	0: playback is completed
filepath	FString	Path of stored recording file

Error codes

Error Code Value	Cause	Suggested Solution
20481	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Parameter is empty.	Check whether the API parameters in the code are correct.
20484	Internal error.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString filepath = JsonObject->GetStringField(TEXT("file_path"));
            onPttPlayFileCompleted(result, filepath);
        }
    }
}
```

Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

API prototype

```
ITMGPTT virtual int StopPlayFile()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->StopPlayFile();
```

Getting audio file size

This API is used to get the size of an audio file.

API prototype

```
ITMGPTT virtual int GetFileSize(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetFileSize(filePath);
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

API prototype

```
ITMGPTT virtual int GetVoiceFileDuration(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetVoiceFileDuration(filePath);
```

Fast Recording-to-Text Conversion

API	Description
SpeechToText	Converts speech to text

Converting audio file to text

This API is used to convert a specified audio file to text.

API prototype

```
ITMGPTT virtual void SpeechToText(const char* fileID)
```

Parameter	Type	Description
fileID	const char*	Audio file URL

Sample code

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(fileID);
```

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

API prototype

```
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage)
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage,cons
```

Parameter	Type	Description
fileID	const char*	The URL of the audio file, which will be retained on the server for 90 days.
speechLanguage	const char*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translatelanguage	const char*	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(filePath,"cmn-Hans-CN","cmn-Hans-C
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` and `text` (recognized text).

Parameter	Type	Description
result	int32	0: recording is completed
fileid	FString	URL of recording file, which will be retained on the server for 90 days
text	FString	Converted text

Error codes

Error Code Value	Cause	Suggested Solution
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.
32784	Incorrect speech-to-text conversion parameter.	Check whether the API parameter `fileid` in the code is empty.
32785	Speech-to-text translation returned an error.	Error with the backend of voice messaging and speech-to-text feature. Analyze logs, get

		the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
        switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
            ...
                    case ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
            {
            // Process
            break;
            }
        }
}
```

Voice Message Volume Level APIs

API	Description
GetMicLevel	Gets real-time mic volume level
SetMicVolume	Sets recording volume level
GetMicVolume	Gets recording volume level
GetSpeakerLevel	Gets real-time speaker volume
SetSpeakerVolume	Sets playback volume level

GetSpeakerVolume	Gets playback volume level

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetMicLevel()
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetMicLevel();
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

API prototype

```
ITMGPTT virtual int SetMicVolume(int vol)
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->SetMicVolume(100);
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetMicVolume()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetMicVolume();
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetSpeakerLevel()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetSpeakerLevel();
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice message. Value range: 0-200.

API prototype

```
ITMGPTT virtual int SetSpeakerVolume(int vol)
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->SetSpeakerVolume(100);
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice message. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetSpeakerVolume()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetSpeakerVolume();
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for SDK usage analysis.

API prototype

```
ITMGContext virtual const char* GetSDKVersion()
```

Sample code

```
ITMGContextGetInstance()->GetSDKVersion();
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
ITMGContext int SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

`ITMG_LOG_LEVEL` description:

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContextGetInstance()->SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

OS	Path
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API prototype

```
ITMGContext virtual int SetLogPath(const char* logDir)
```

Parameter	Type	Description
logDir	const char*	Path

Sample code

```
cosnt char* logDir = ""// Set a path by yourself
ITMGContext* context = ITMGContextGetInstance();
context->SetLogPath(logDir);
```

Callback Messages

Message	Description	Parameter
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected for network	result; error

	or other reasons	
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	The reconnection to the room started	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	The reconnection to the room succeeded	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status was changed	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker device was changed	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A new speaker device was added	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker device was	result; error

	lost	
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A new mic device was added	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic device was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic device was changed	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room quality changed	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Voice message recording was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Voice message upload was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Voice message download was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Voice message playback was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast speech-to-text conversion was completed	result; text;file_id

ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	Streaming speech-to-text conversion is in progress	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed	result; text;file_id

SDK for Windows

Project Configuration

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to configure a Windows project for GME APIs for Windows.

SDK Preparations

1. Download the applicable demo and SDK. For more information, please see Download Guide.
2. Decompress the obtained SDK resources.
3. The folder contains:

include: GME SDK header files.

libs: GME SDK dll files.

Configuration Guide

1. Copy the `include` and `libs` folder into the project folder.
2. On the project's property page, add the SDK file address to the `additional include` directory.

Voice Chat

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to access and debug GME client APIs for the voice chat feature for Windows.

Key Considerations for Using GME

GME provides the real-time voice, voice message, and speech-to-text conversion services, which all depend on core APIs such as `Init` and `Poll` .

Key notes

You have created a GME application and obtained the `AppID` and `Key` of the SDK as instructed in Activating Services.

You have **activated the real-time voice, voice message, and speech-to-text services of GME** as instructed in Activating Services.

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

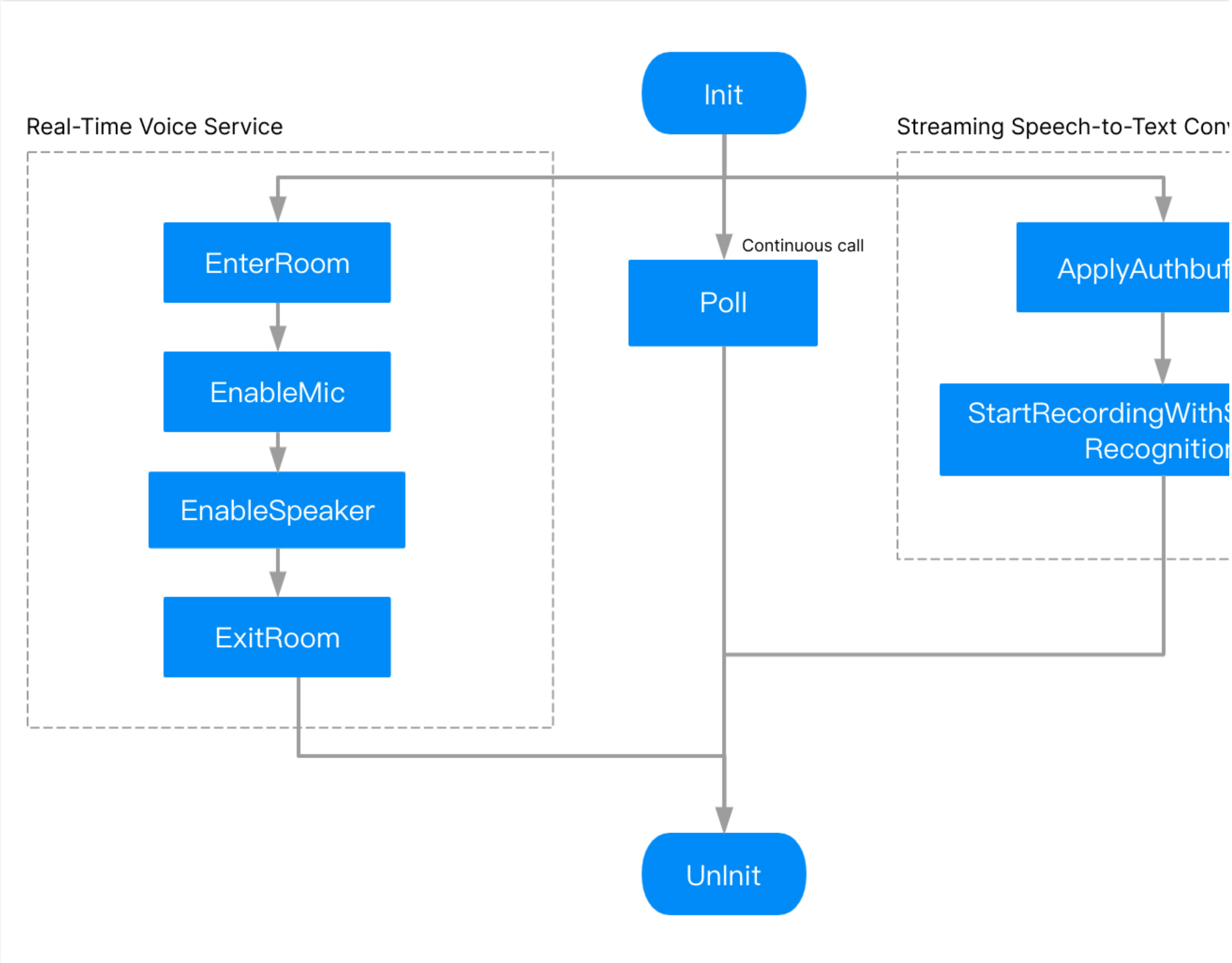
The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error code, please see Error Codes.

Connecting to the SDK

Directions

Key processes involved in SDK connection are as follows:



1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Entering a voice chat room, API: EnterRoom
4. Enabling the microphone, API: EnableMic
5. Enabling the speaker, API: EnableSpeaker
6. Exiting a voice room, API: ExitRoom
7. Uninitializing GME, API: UnInit

C++ classes

Class	Description
ITMGContext	Key APIs
ITMGDelegate	Callback APIs
ITMGRoom	Room APIs

ITMGRoomManager	Room management APIs as described in Integrating GME Chat Room Management
ITMGAudioCtrl	Audio APIs
ITMGAudioEffectCtrl	Sound effect and accompaniment APIs
ITMGPTT	Voice message and speech-to-text conversion APIs

Key APIs

API	Description
Init	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME

Imported header files

You need to import the header file `tmg_sdk.h` first before you can access GME. The classes in the header file inherit `ITMGDelegate` for message delivery and callback.

Sample code

```
#include "auth_buffer.h"
#include "tmg_sdk.h"
#include "AdvanceHeaders/tmg_sdk_adv.h"
#include <vector>
```

Callback

Setting callback sample

```
// When initializing the SDK
m_pTmgContext = ITMGContextGetInstance();
m_pTmgContext->SetTMGDelegate(this);

// In the destructor
CTMGSDK_For_AudioDlg::~CTMGSDK_For_AudioDlg()
```



```
{
              ITMGContextGetInstance()->SetTMGDelegate(NULL);
}
```

Message delivery

The API class uses the `Delegate` method to send callback notifications to the application.
`ITMG_MAIN_EVENT_TYPE` indicates the message type. The data on Windows is in json string format. For the key-value pairs, please see the relevant documentation.

```
// Declaration in the header file
virtual void OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data);
// Sample code
void CTMGSDK_For_AudioDlg::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data
{
              switch(eventType)
              {
              case ITMG_MAIN_EVENT_TYPE_XXXX_XXXX:
                   {
                       // Process the callback
                   }
                   break;
              }
}
```

Getting singleton

The GME SDK is provided in the form of a singleton, all calls begin with `ITMGContext`, and callbacks are passed to the application through `ITMGDelegate`, which should be configured first.

Sample code

```
ITMGContext* m_pTmgContext;
m_pTmgContext->Init(AppID, OpenID);
```

Initializing SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

Parameter	Type	Description
sdkAppId	const char*	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	const char*	`openID` can only be in `Int64` type, which is passed in after being converted to a `const char*` . You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
AV_OK = 0	Initialized SDK successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

Notes on 7015 error code

The 7015 error code is judged by md5. If this error is reported during integration, please check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party reinforcement, Unity packaging mechanism and other factors, the md5 of the library file will be affected, resulting in misjudgment. **Please ignore this error in the logic for official release**, and try to avoid displaying it in the UI.

Sample code

```
#define SDKAPPID3RD "14000xxxxx"
cosnt char* openId="10001";
ITMGContext* context = ITMGContextGetInstance();
context->Init(SDKAPPID3RD, openId);
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update` . `Poll` is the message pump of GME, and the `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire

SDK service will run exceptionally.

You can refer to the `EnginePollHelper.cpp` file in the demo.

Calling the `Poll` API periodically

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```

Sample code

```
// Declaration in the header file

// Code implementation
void TMGTestScene::update(float delta)
{
    ITMGContextGetInstance()->Poll();
}
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
ITMGContext int Pause()
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
ITMGContext int Resume()
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

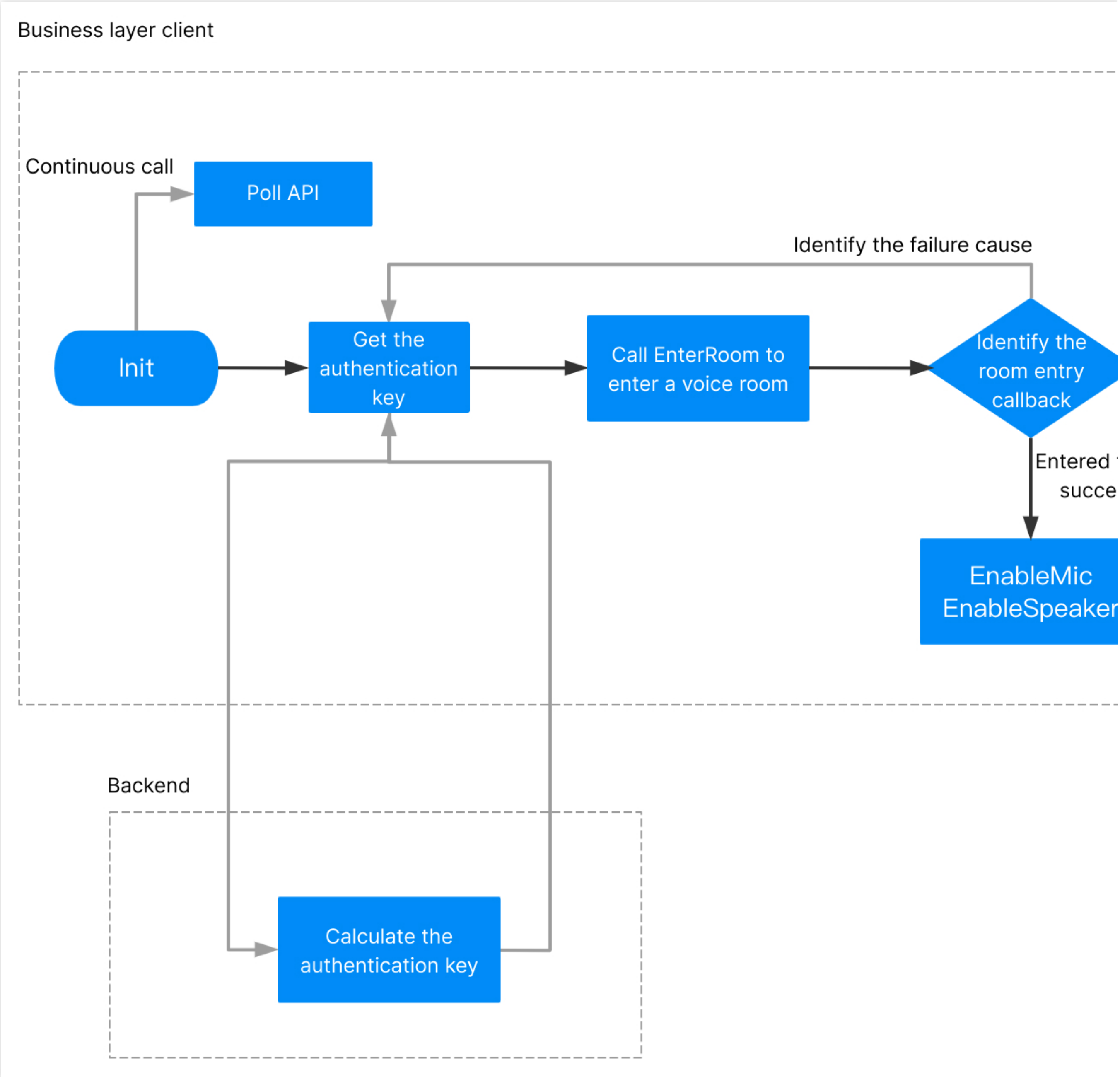
API prototype

```
ITMGContext int Uninit()
```

Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, please see FAQs About Voice Chat.



API	Description
GenAuthBuffer	Calculates the local authentication key
EnterRoom	Enters a room
ExitRoom	Exits the room
IsRoomEntered	Determines whether room entry is successful
SwitchRoom	Switches the room quickly

Local authentication key calculation

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.

API prototype

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

Parameter	Type	Description
dwSdkAppID	unsigned int	`AppId` from the Tencent Cloud console
strRoomID	const char*	Room ID, which can contain up to 127 characters.
strOpenID	const char*	User ID, which is the same as `openID` during initialization.
strKey	const char*	Permission key from the Tencent Cloud console
strAuthBuffer	const char*	Returned `authbuff`
bufferLength	int	Length of the `authbuff` passed in. 500 is recommended.

Sample code

```
unsigned int bufferLen = 512;
unsigned char retAuthBuff[512] = {0};
QAVSDK_AuthBuffer_GenAuthBuffer(atoi(SDKAPPID3RD), roomId, "10001", AUTHKEY,retAuth
```

Entering a room

This API is used to enter a room with the generated authentication information. The mic and speaker are not enabled by default after room entry.

Note:

If the room entry callback result is `0`, the room entry is successful. If `0` is returned from the `EnterRoom` API, it doesn't necessarily mean that the room entry is successful.

The audio type of the room is determined by the first user entering the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will change to the smooth sound quality. Only after a member in the room calls the `ChangeRoomType` API will the audio type of the room be changed.

API prototype

```
ITMGContext virtual int EnterRoom(const char*  roomID, ITMG_ROOM_TYPE roomType, con
```

Parameter	Type	Description
roomID	const char*	Room ID, which can contain up to 127 characters.
roomType	ITMG_ROOM_TYPE	Room type. We recommend you select `ITMG_ROOM_TYPE_FLUENCY` for games. For more information on room audio types, see Sound Quality.
authBuffer	const char*	Authentication key
buffLen	int	Authentication key length

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->EnterRoom(roomID, ITMG_ROOM_TYPE_FLUENCY, (char*)retAuthBuff,bufferLen);
```

Callback for room entry

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function for callback and processing. A successful callback means that the room entry is successful, and the billing starts.

Billing references

Purchase Guide

Billing FAQs

Will Voice Chat still be charged when client is offlined?

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
                    switch (eventType) {
                        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
                        {
                            ListMicDevices();
                            ListSpeakerDevices();

                            std::string strText = "EnterRoom complete: ret=";
                            strText += data;
                            m_EditMonitor.SetWindowText(MByteToWChar(strText).c_str
                            }
                    }
```



```
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnected callback prompt `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` . At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` . When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` .

Error codes

Error Code Value	Cause and Suggested Solution
7006	Authentication failed: The `AppID` does not exist or is incorrect. An error occurred while authenticating the `authbuff` . Authentication expired. The `OpenId` does not meet the specification.
7007	Already in another room.
1001	The user was already in the process of entering a room but repeated this operation. It is recommended not to call the room entering API until the room entry callback is returned.
1003	The user was already in the room and called the room entering API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally.

Exiting a room

This API is used to exit the current room. It is an async API. The returned value `AV_OK` indicates a successful async delivery. If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API; instead, you can directly call the `EnterRoom` API.

API prototype

```
ITMGContext virtual int ExitRoom()
```

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->ExitRoom();
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM`.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
        // Process
        break;
        }
    }
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}

Determining whether user has entered room

This API is used to determine whether the user has entered a room. A value in bool type will be returned. Do not call this API during room entry.

API prototype

```
ITMGContext virtual bool IsRoomEntered()
```

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
context->IsRoomEntered();
```

Switching room

User can call this API to quickly switch the voice chat room after entering the room. After the room is switched, the device is not reset, that is, if the microphone is already enabled in this room, the microphone will keep enabled after the room is switched.

The callback for quickly switching rooms is `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM` , and the fields are `error_info` and `result` .

API prototype

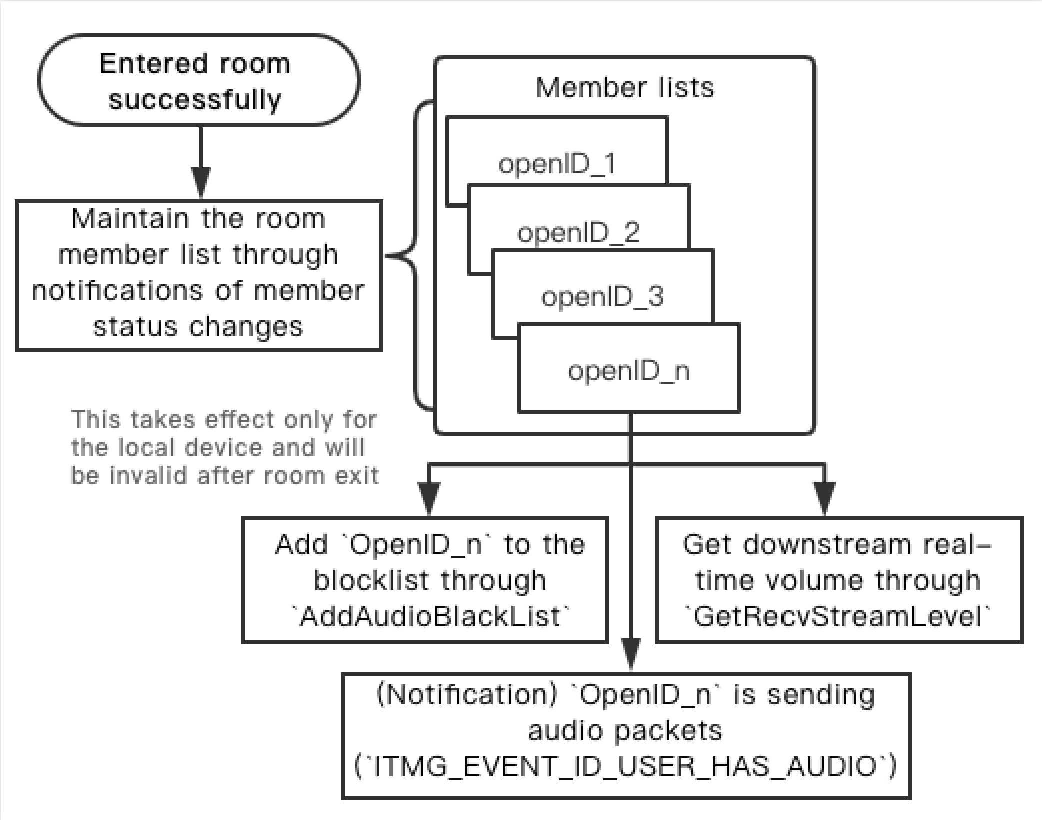
```
ITMGContext virtual int SwitchRoom(const char* targetRoomID, const char* authBuff,
```

Type descriptions

Parameter	Type	Description
targetRoomID	const char*	ID of the room to enter
authBuffer	const char*	Generates a new authentication key with the ID of the room to enter
buffLen	int	Authentication key length

Room Status Maintenance

APIs in this section are used to display speaking members and members entering or exiting the room and mute a member in the room at the business layer.



API/Notification	Description
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The member status changed
AddAudioBlackList	Mutes a member in the room
RemoveAudioBlackList	Unmutes a user

Notification events of member room entry and speaking status

This API is used to obtain the user speaking in the room and display it in the UI, and to send a notification when someone entering or exiting the room.

Notification for this event will be sent only when the status changes. To get member status in real time, cache the notification when it is received at a higher layer. The event message is `ITMG_MAIN_EVNET_TYPE_USER_UPDATE` , where the data contains `event_id` and `user_list` . The event message will be identified in the `OnEvent` function.

Notifications for audio events are subject to a threshold, and a notification will be sent only when this threshold is exceeded. The notification "A member has stopped sending audio packets" will be sent if no audio packets are received in more than two seconds.

event_id	Description	Maintenance
ITMG_EVENT_ID_USER_ENTER	Return the `openid` of the member entering the	Member list

	room.	
ITMG_EVENT_ID_USER_EXIT	Return the `openid` of the member exiting the room.	Member list
ITMG_EVENT_ID_USER_HAS_AUDIO	Return the `openid` of the member sending audio packets in the room. This event can be used to determine whether a user is speaking and display the voiceprint effect.	Chat member list
ITMG_EVENT_ID_USER_NO_AUDIO	Return the `openid` of the member stopping sending audio packets in the room.	Chat member list

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
            case ITMG_MAIN_EVNET_TYPE_USER_UPDATE:
        {
        // Process
        // Parse the parameter to get `eventID` and `user_list`
            switch (eventID)
             {
             case ITMG_EVENT_ID_USER_ENTER:
                  // A member enters the room
                  break;
             case ITMG_EVENT_ID_USER_EXIT:
                  // A member exits the room
                break;
            case ITMG_EVENT_ID_USER_HAS_AUDIO:
                // A member sends audio packets
                break;
            case ITMG_EVENT_ID_USER_NO_AUDIO:
                // A member stops sending audio packets
                break;
             default:
                break;
            }
        break;
        }
    }
}
```

Muting a member in the room

This API is used to add an ID to the audio data blocklist. This operation blocks audio from someone and only applies to the local device without affecting other devices. The returned value `0` indicates that the call is successful.

Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B;

If B blocks neither A nor C, B can hear both of them;

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

API prototype

```
ITMGContext ITMGAudioCtrl int AddAudioBlackList(const char* openId)
```

Parameter	Type	Description
openId	char*	`openid` of the user to be blocked

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->AddAudioBlackList(openId);
```

Unmuting

This API is used to remove an ID from the audio data blacklist. A returned value of 0 indicates the call is successful.

API prototype

```
ITMGContext ITMGAudioCtrl int RemoveAudioBlackList(const char* openId)
```

Parameter	Type	Description
openId	char*	ID to be unblocked

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->RemoveAudioBlackList(openId);
```

Voice Chat Capturing APIs

The voice chat APIs can only be called after SDK initialization and room entry.

When the user clicks the button of enabling/disabling the mic or speaker on the UI, we recommend you call the `EnableMic` or `EnableSpeaker` API.
To enable the user to press the mic button on the UI to speak and release it to stop speaking, we recommend you call `EnableAudioCaptureDevice` once during room entry and call `EnableAudioSend` to enable the user to speak while pressing the button.

API	Description
EnableMic	Enables/Disables the mic
GetMicState	Gets the mic status
EnableAudioCaptureDevice	Enables/Disables the capturing device
IsAudioCaptureDeviceEnabled	Gets the capturing device status
EnableAudioSend	Enables/Disables audio upstreaming
IsAudioSendEnabled	Gets the audio upstreaming status
GetMicLevel	Gets the real-time mic volume level
GetSendStreamLevel	Gets real-time audio upstreaming volume
SetMicVolume	Sets the mic volume level
GetMicVolume	Gets the mic volume level

Enabling or disabling mic

This API is used to enable/disable the mic. The mic and speaker are not enabled by default after room entry.
EnableMic = EnableAudioCaptureDevice + EnableAudioSend

API prototype

```
ITMGAudioCtrl virtual int EnableMic(bool bEnabled)
```

Parameter	Type	Description
bEnabled	bool	To enable the mic, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable mic
ITMGContextGetInstance()->GetAudioCtrl()->EnableMic(true);
```

Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

API prototype

```
ITMGAudioCtrl virtual int GetMicState()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicState();
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.
This API can only be called after room entry. The device will be disabled automatically after room exit.
Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

API prototype

```
ITMGAudioCtrl virtual int EnableAudioCaptureDevice(bool enable)
```

Parameter	Type	Description
enable	bool	To enable the capturing device, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable capturing device
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioCaptureDevice(true);
```

Getting the capturing device status

This API is used to get the status of a capturing device.

API prototype

```
ITMGContext virtual bool IsAudioCaptureDeviceEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioCaptureDeviceEnabled();
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain mute. For more information on how to enable/disable the capturing device, please see the `EnableAudioCaptureDevice` API.

API prototype

```
ITMGContext  virtual int EnableAudioSend(bool bEnable)
```

Parameter	Type	Description
bEnable	bool	To enable audio upstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioSend(true);
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

API prototype

```
ITMGContext virtual bool IsAudioSendEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioSendEnabled();
```

Getting the real-time mic volume

This API is used to get the real-time mic volume. An int-type value in the range of 0-100 will be returned. It is recommended to call this API once every 20 ms.

API prototype

```
ITMGAudioCtrl virtual int GetMicLevel()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicLevel();
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume. An int-type value in the range of 0-100 will be returned.

API prototype

```
ITMGAudioCtrl virtual int GetSendStreamLevel()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSendStreamLevel();
```

Setting the mic software volume

This API is used to set the mic volume level. The corresponding parameter is `volume` , which is equivalent to attenuating or gaining the captured sound.

API prototype

```
ITMGAudioCtrl virtual int SetMicVolume(int vol)
```

Parameter	Type	Description
vol	int	Value range: 0–200. Default value: 100. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
int micVol = (int)(value * 100);
ITMGContextGetInstance()->GetAudioCtrl()->SetMicVolume(vol);
```

Getting the mic software volume

This API is used to obtain the microphone volume. An "int" value is returned. Value 101 represents API SetMicVolume has not been called.

API prototype

```
ITMGAudioCtrl virtual int GetMicVolume()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetMicVolume();
```

Voice Chat Playback APIs

API	Description
EnableSpeaker	Enables/Disables the speaker
GetSpeakerState	Gets the speaker status
EnableAudioPlayDevice	Enables/Disables the playback device
IsAudioPlayDeviceEnabled	Gets playback device status
EnableAudioRecv	Enables/Disables audio downstreaming
IsAudioRecvEnabled	Gets the audio downstreaming status
GetSpeakerLevel	Gets the real-time speaker volume level
GetRecvStreamLevel	Gets the real-time downstreaming audio volume levels of other members in the room
SetSpeakerVolume	Sets the speaker volume level
GetSpeakerVolume	Gets the speaker volume level

Enabling or disabling speaker

This API is used to enable/disable the speaker. **EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv**

API prototype

```
ITMGAudioCtrl virtual int EnableSpeaker(bool enable)
```

Parameter	Type	Description
enable	bool	To disable the speaker, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
// Enable the speaker
ITMGContextGetInstance()->GetAudioCtrl()->EnableSpeaker(true);
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, and 1 is on.

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerState()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerState();
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

API prototype

```
ITMGAudioCtrl virtual int EnableAudioPlayDevice(bool enable)
```

Parameter	Type	Description
enable	bool	To disable the playback device, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioPlayDevice(true);
```

Getting the playback device status

This API is used to get the status of a playback device.

API prototype

```
ITMGAudioCtrl virtual bool IsAudioPlayDeviceEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioPlayDeviceEnabled();
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain mute. For more information on how to enable/disable the playback device, please see the `EnableAudioPlayDevice` API.

API prototype

```
ITMGAudioCtrl virtual int EnableAudioRecv(bool enable)
```

Parameter	Type	Description
enable	bool	To enable audio downstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableAudioRecv(true);
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

API prototype

```
ITMGAudioCtrl virtual bool IsAudioRecvEnabled()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->IsAudioRecvEnabled();
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume. An int-type value will be returned to indicate the volume. It is recommended to call this API once every 20 ms.

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerLevel()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerLevel();
```

Getting the real-time downstreaming audio levels of other members in room

This API is used to get the real-time audio downstreaming volume of other members in the room. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGAudioCtrl virtual int GetRecvStreamLevel(const char* openId)
```

Parameter	Type	Description
openId	char*	`openId` of other members in the room

Sample code

```
iter->second.level = ITMGContextGetInstance()->GetAudioCtrl()->GetRecvStreamLevel(i
```

Dynamically setting the volume of a member of the room

This API is used to set the volume of a member in the room. It takes effect only on the local.

API prototype

```
ITMGAudioCtrl virtual int SetSpeakerVolumeByOpenID(const char* openId, int vol) = 0
```

Parameter	Type	Description
openId	const char*	`OpenID` of the user whose volume level needs to be set
vol	int	Percentage. Recommended value range: 0–200. Default value: `100` .

Getting volume percentage

This API is used to get the volume level set by `SetSpeakerVolumeByOpenID` .

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerVolumeByOpenID(const char* openId) = 0;
```

Parameter	Type	Description
openId	const char*	`OpenID` of the user whose volume level needs to be set

Returned values

API returns volume percentage set by OpenID, where 100 is by default.

Setting the speaker volume

This API is used to set the speaker volume.

API prototype

```
ITMGAudioCtrl virtual int SetSpeakerVolume(int vol)
```

Parameter	Type	Description
vol	int	Volume level. Value range: 0–200. Default value: 100. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
int vol = 100;
ITMGContextGetInstance()->GetAudioCtrl()->SetSpeakerVolume(vol);
```

Getting the speaker volume

This API is used to get the speaker volume. An int-type value will be returned to indicate the volume. 101 indicates that the `SetSpeakerVolume` API has not been called.

"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

API prototype

```
ITMGAudioCtrl virtual int GetSpeakerVolume()
```

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSpeakerVolume();
```

Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

API prototype

```
ITMGAudioCtrl virtual int EnableLoopBack(bool enable)
```

Parameter	Type	Description
enable	bool	Specifies whether to enable

Sample code

```
ITMGContextGetInstance()->GetAudioCtrl()->EnableLoopBack(true);
```

Getting user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, please see the `EnterRoom` API.

API prototype

```
class ITMGRoom {
public:
    virtual ~ITMGRoom() {} ;
    virtual int GetRoomType() = 0;

};
```

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
ITMGContextGetInstance()->GetRoom()->GetRoomType();
```

Getting the room ID

This API is used to get the voice chat room ID and can be called only after a successful room entry.

API prototype

```
ITMGRoom virtual int GetRoomID(char* pBuffer, int nLength) = 0;
```

Parameter	Type	Description
pBuffer	char*	It is used to receive the returned `roomid` .
nLength	int	`pBuffer` length. Value range: 128–256.

Modifying user's room audio type

This API is used to modify a user's room audio type. For the result, please see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

API prototype

```
IITMGContext TMGRoom public int ChangeRoomType((ITMG_ROOM_TYPE roomType)
```

Parameter	Type	Description
roomType	ITMG_ROOM_TYPE	Room type to be switched to the target type. For room audio types, please see the `EnterRoom` API.

Sample code

```
ITMGContext* context = ITMGContextGetInstance();
ITMGContextGetInstance()->GetRoom()->ChangeRoomType(ITMG_ROOM_TYPE_FLUENCY);
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; new_room_type	{"error_info":"","new_room_type":0

Callback for modifying the room type

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result` , `error_info` , and `new_room_type` . The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	The existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	A user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	A user is already in the room, and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	A room member calls the `ChangeRoomType` API to request a change of room audio type.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE == t
        {
        // Process room type events
     }
}
```

The monitoring event of room call quality

This is the quality monitoring event, which is triggered once every two seconds after room entry, and its message is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY` . The returned parameters include `weight` , `loss` , and `delay` , which are as detailed below:

Parameter	Type	Description
weight	int	Value range: 1-50. 50 indicates excellent sound quality, 1 indicates very poor (barely usable) sound quality, and 0 represents an initial meaningless value. Generally, if the value is below 30, the business layer will remind users that the network is poor and recommend them to switch the network.
loss	double	Upstream packet loss rate
delay	int	Voice chat delay in ms

Getting version number

This API is used to get the SDK version number for analysis.

API prototype

```
ITMGContext virtual const char* GetSDKVersion()
```

Sample code

```
ITMGContextGetInstance()->GetSDKVersion();
```

Checking mic status

Function prototype

```
ITMGContext virtual ITMG_CHECK_MIC_STATUS CheckMic() = 0;
```

Returned value handling

Returned Value	Description	Handling
ITMG_CHECK_MIC_STATUS_AVAILABLE = 0	Normally available	No handling required
ITMG_CHECK_MIC_STATUS_NO_GRANTED = 2	Access not obtained/denied	The access permission needs to be obtained before the mic is enabled
ITMG_CHECK_MIC_STATUS_INVALID_MIC = 3	No device available	Generally, this error will be reported on PCs when no mics are available. Prompt the user to insert a headset or mic
ITMG_CHECK_MIC_STATUS_NOT_INIT = 5	Not initialized	Call `EnableMic` after `Init`

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
ITMGContext int SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

`ITMG_LOG_LEVEL` is as detailed below:

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContextGetInstance()->SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

OS	Path
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API prototype

```
ITMGContext virtual int SetLogPath(const char* logDir)
```

Parameter	Type	Description
logDir	const char*	Path

Sample code

```
cosnt char* logDir = ""// Set a path by yourself

ITMGContext* context = ITMGContextGetInstance();
context->SetLogPath(logDir);
```

Getting the printed log path

This API is used to get the log path. Its returned value is a string of `const char*` type.

API prototype

```
ITMGContext virtual const char* GetLogPath() = 0;
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

API prototype

```
ITMGRoom virtual const char* GetQualityTips()
```

Sample code

```
ITMGContextGetInstance()->GetRoom()->GetQualityTips();
```

Callback Messages

Message	Description	Parameter
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room	result; error

ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected for network or other reasons	result; error
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The room members were updated	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	Room reconnection started	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	Room reconnection succeeded	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status changed	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker was changed	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A speaker was added	result; error

ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A mic was added	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic was changed	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	Room quality message	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Recording of a voice message was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Upload of a voice message was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Download of a voice message was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Playback of a voice message was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast recording-to-	result; text;file_id

	text conversion was completed	
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	A voice message is being converted into text in a streaming manner	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed	result; text;file_id

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to integrate with and debug GME client APIs for the voice messaging and speech-to-text services for Windows.

Key Considerations for Using GME

GME provides the real-time voice and voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

Notes

You have created a GME application and obtained the SDK AppID and key. For more information, see Activating Services.
You have activated **GME real-time voice and voice messaging and speech-to-text services**. For more information, see Activating Services.
Configure your project before using GME; otherwise, the SDK will not take effect.
After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.
GME APIs should be called in the same thread.
The `Poll` API should be called periodically for GME to trigger event callbacks.
For detailed error code, see Error Codes.

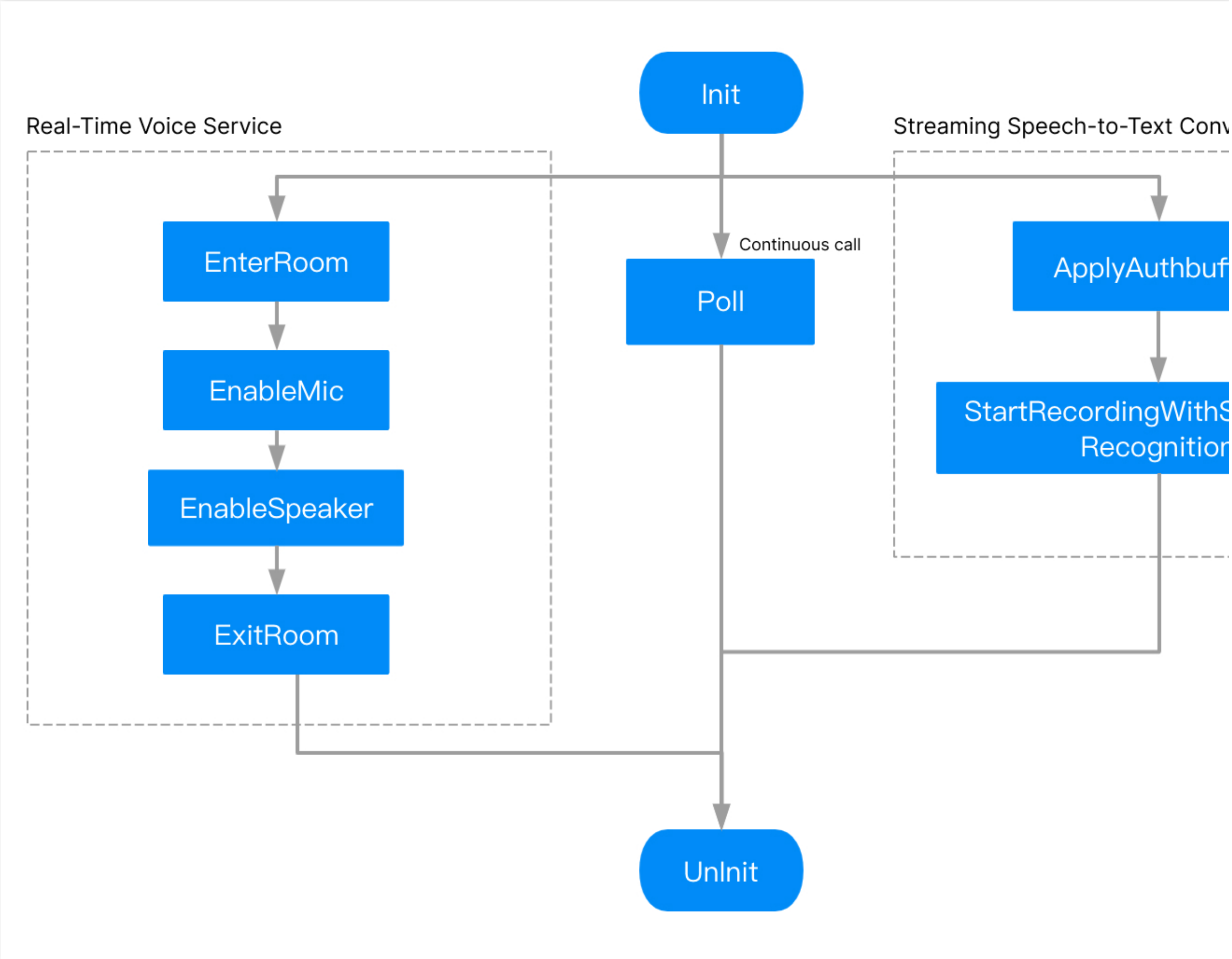
Note:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.
Non-streaming speech-to-text API ***SpeechToText()***: There can be up to 10 concurrent requests per account.
Streaming speech-to-text API ***StartRecordingWithStreamingRecognition()***: There can be up to 50 concurrent requests per account.
Real-time streaming speech-to-text API ***StartRealTimeASR()***: There can be up to 50 concurrent requests per account.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Initializing authentication, API: ApplyPTTAuthbuffer
4. Starting streaming speech recognition, API: StartRecordingWithStreamingRecognition
5. Stop recording, API: StopRecording
6. Uninitializing GME, API: UnInit

C++ classes

Class	Description
ITMGContext	Key APIs
ITMGPTT	Voice messaging and speech-to-text APIs

Key APIs

API	Description
Init	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME

Importing the header file

```
#include "auth_buffer.h"
#include "tmg_sdk.h"
#include "AdvanceHeaders/tmg_sdk_adv.h"
#include <vector>
```

Callback

Setting callback sample

```
// When initializing the SDK
m_pTmgContext = ITMGContextGetInstance();
m_pTmgContext->SetTMGDelegate(this);

// In the destructor
CTMGSDK_For_AudioDlg::~CTMGSDK_For_AudioDlg()
{
            ITMGContextGetInstance()->SetTMGDelegate(NULL);
}
```

Message delivery

The API class uses the `Delegate` method to send callback notifications to the application. `ITMG_MAIN_EVENT_TYPE` indicates the message type. The data on Windows is in json string format. For the key-value pairs, please see the relevant documentation.

```
// Declaration in the header file
virtual void OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data);
```

```
// Sample code
void CTMGSDK_For_AudioDlg::OnEvent(ITMG_MAIN_EVENT_TYPE eventType, const char* data
{
            switch(eventType)
            {
            case ITMG_MAIN_EVENT_TYPE_XXXX_XXXX:
                {
                    // Process the callback
                }
                break;
            }
}
```

Getting singleton

The GME SDK is provided in the form of a singleton, all calls begin with `ITMGContext`, and callbacks are passed to the application through `ITMGDelegate`, which should be configured first.

Sample code

```
ITMGContext* m_pTmgContext;
m_pTmgContext->Init(AppID, OpenID);
```

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
ITMGContext virtual int Init(const char* sdkAppId, const char* openId)
```

Parameter	Type	Description
sdkAppId	const char*	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	const char*	`openID` can only be in `Int64` type, which is passed in after being converted to a `const char*`. You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
AV_OK = 0	Initialized SDK successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

Notes on 7015 error code

The 7015 error code is judged by md5. If this error is reported during integration, please check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party reinforcement, Unity packaging mechanism and other factors, the md5 of the library file will be affected, resulting in misjudgment. **Please ignore this error in the logic for official release**, and try to avoid displaying it in the UI.

Sample code

```
#define SDKAPPID3RD "14000xxxxxx"
cosnt char* openId="10001";
ITMGContext* context = ITMGContextGetInstance();
context->Init(SDKAPPID3RD, openId);
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update` . `Poll` is the message pump of GME, and the `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.

You can refer to the `EnginePollHelper.cpp` file in the demo.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
class ITMGContext {
protected:
    virtual ~ITMGContext() {}

public:
    virtual void Poll()= 0;
}
```

Sample code

```
void TMGTestScene::update(float delta)
{
    ITMGContextGetInstance()->Poll();
}
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
ITMGContext int Pause()
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
ITMGContext int Resume()
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

API prototype

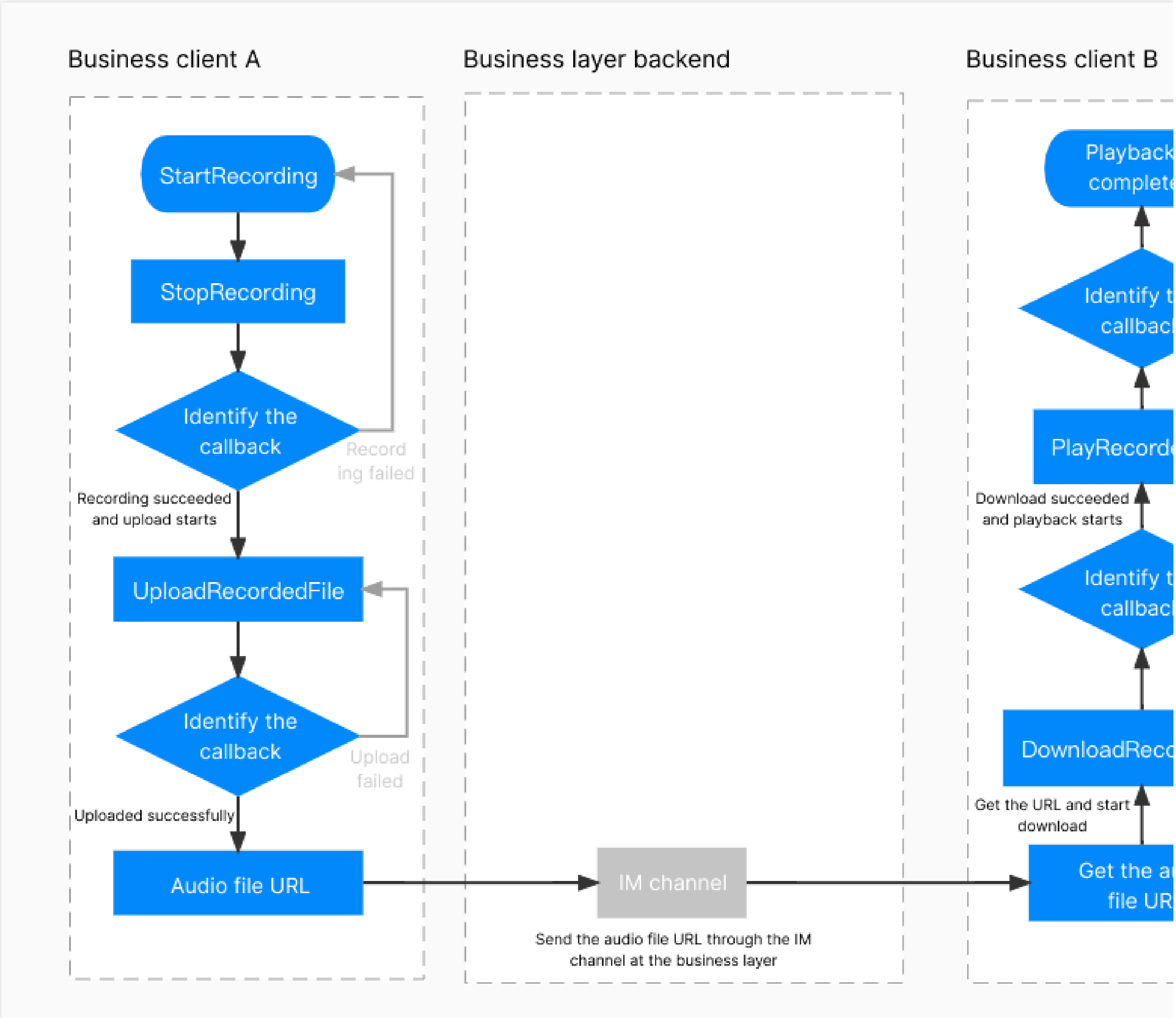
```
ITMGContext int Uninit()
```

Voice Messaging and Speech-to-Text Services

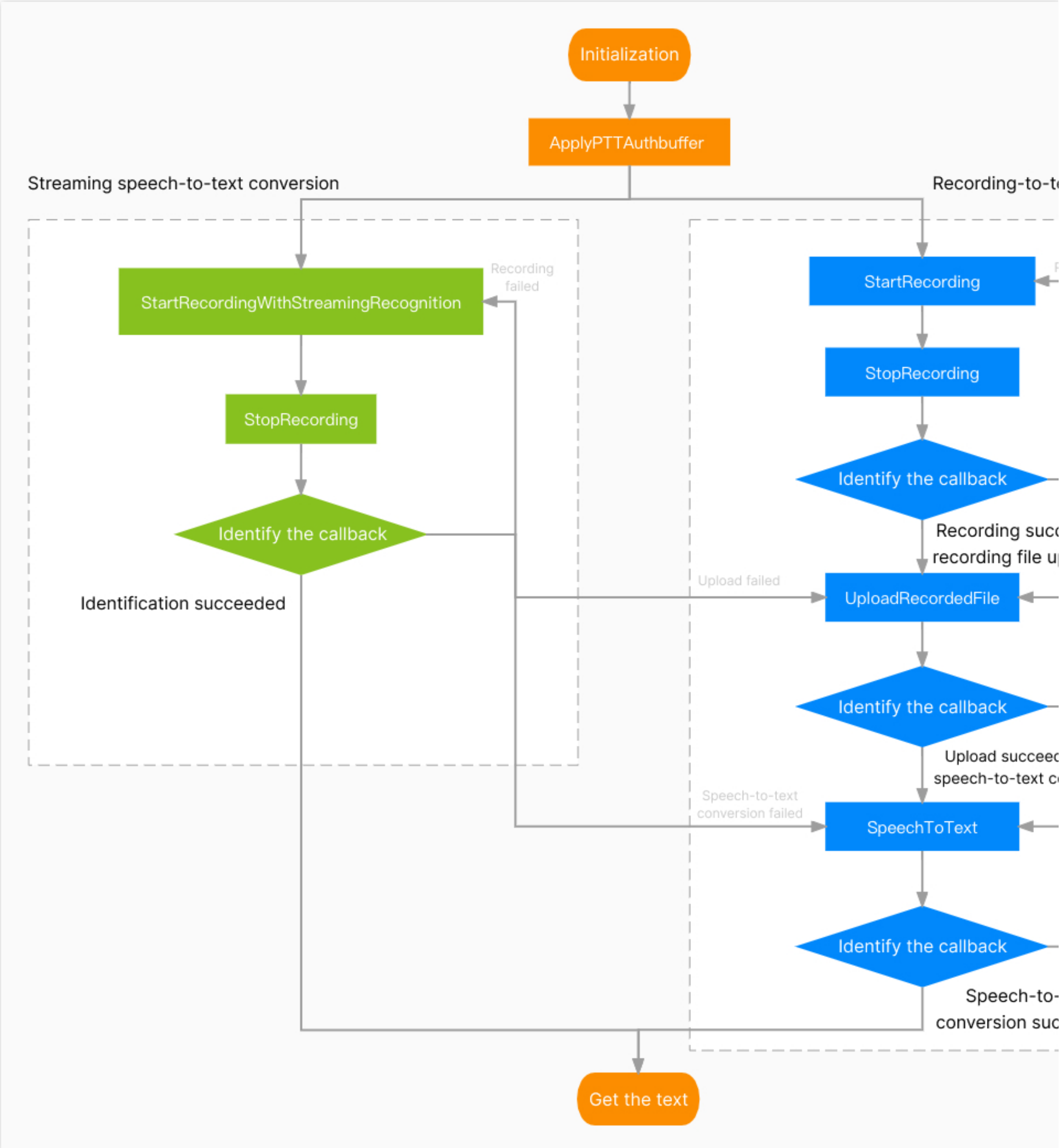
Note:

The speech-to-text service consists of fast recording-to-text conversion and streaming speech-to-text conversion. You do not need to enter a voice chat room when using the voice message service.

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, call the `SetMaxMessageLength` API to set it after initialization.



Flowchart for using the speech-to-text service



API	Description
GenAuthBuffer	Generates the local authentication key
ApplyPTTAuthbuffer	Initializes authentication
SetMaxMessageLength	Specifies the maximum length of voice message

Generating the local authentication key

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.

API prototype

```
int  QAVSDK_AuthBuffer_GenAuthBuffer(unsigned int dwSdkAppID, const char* strRoomID
    const char* strKey, unsigned char* strAuthBuffer, unsigned int bufferLength);
```

Parameter	Type	Description
dwSdkAppID	int	`AppId` from the Tencent Cloud console.
strRoomID	const char*	Enter `null` or an empty string
strOpenID	const char*	User ID, which is the same as `openID` during initialization.
strKey	const char*	Permission key from the Tencent Cloud console.
strAuthBuffer	const char*	Returned `authbuff` .
bufferLength	int	Length of the `authbuff` passed in. 500 is recommended.

Application authentication

After the authentication information is generated, the authentication is assigned to the SDK.

API prototype

```
ITMGPTT virtual int ApplyPTTAuthbuffer(const char* authBuffer, int authBufferLen)
```

Parameter	Type	Description
authBuffer	const char*	Authentication
authBufferLen	int	Authentication length

Sample code

```
ITMGContextGetInstance()->GetPTT()->ApplyPTTAuthbuffer(authBuffer,authBufferLen);
```

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

API prototype

```
ITMGPTT virtual int SetMaxMessageLength(int msTime)
```

Parameter	Type	Description
msTime	int	Audio duration in ms. Value range: 1000 < msTime <= 58000

Sample code

```
int msTime = 10000;
ITMGContextGetInstance()->GetPTT()->SetMaxMessageLength(msTime);
```

Streaming Speech Recognition

Voice message and speech-to-text APIs

API	Description
StartRecordingWithStreamingRecognition	Starts streaming recording
StopRecording	Stops recording

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call** Stop recording.

API prototype

```
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath)
ITMGPTT virtual int StartRecordingWithStreamingRecognition(const char* filePath,con
```

Parameter	Type	Description
filePath	const char*	Path of stored audio file
speechLanguage	const char*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.

translateLanguage	const char*	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContextGetInstance()->GetPTT()->StartRecordingWithStreamingRecognition(filePath
```

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen on callback messages in the `OnEvent` notification, which is as detailed below:

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real-time during the recording, which is equivalent to returning the recognized text while speaking.

The event message will be identified in the `OnEvent` notification based on the actual needs. The passed parameters include the following four messages.

Message Name	Description
result	Return code indicating whether streaming speech recognition is successful
text	Text converted from speech
file_path	Local path of stored recording file
file_id	Backend URL address of recording file, which will be retained for 90 days

Note:

The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Suggested Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the

		`SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

If the error code 4098 is reported, see FAQs for solutions.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
                switch (eventType) {
                    case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
                    {
                        HandleSTREAM2TEXTComplete(data,true);
                        break;
                        }
                    ...
                            case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_R
                    {
                        HandleSTREAM2TEXTComplete(data, false);
                        break;
                    }
                }
}

void CTMGSDK_For_AudioDlg::HandleSTREAM2TEXTComplete(const char* data, bool isCompl
{
                    std::string strText = "STREAM2TEXT: ret=";
                    strText += data;
                    m_EditMonitor.SetWindowText(MByteToWChar(strText).c_str());
                    Json::Reader reader;
                    Json::Value root;
                    bool parseRet = reader.parse(data, root);
```



```
                    if (!parseRet) {
                        ::SetWindowText(m_EditInfo.GetSafeHwnd(),MByteToWChar(std::
                    }
                    else
                    {
                        if (isComplete) {
                            ::SetWindowText(m_EditUpload.GetSafeHwnd(), MByteToWCha
                        }
                        else {
                            std::string isruning = "STREAMINGRECOGNITION_IS_RUNNING
                            ::SetWindowText(m_EditUpload.GetSafeHwnd(), MByteToWCha
                        }
                    }
}
```

Voice Message Recording

The recording process is as follows: start recording > stop recording > return recording callback > start the next recording.

Voice message and speech-to-text APIs

API	Description
StartRecording	Starts recording
PauseRecording	Pauses recording
ResumeRecording	Resumes recording
StopRecording	Stops recording
CancelRecording	Cancels recording

Starting recording

This API is used to start recording.

API prototype

```
ITMGPTT virtual int StartRecording(const char* fileDir)
```

Parameter	Type	Description

fileDir	const char*	Path of stored audio file

Sample code

```
char buffer[256]={0};
snprintf(buffer, sizeof(buffer), "%sunreal_ptt_local.file", getFilePath().c_str());
ITMGContextGetInstance()->GetPTT()->StartRecording(buffer);
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

API prototype

```
ITMGPTT virtual int StopRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->StopRecording();
```

Callback for recording start

The recording start result will be returned through the callback.

To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file, which must be accessible and cannot be the `fileid`

Error codes

Error Code Value	Cause	Suggested Solution
4097	Parameter is empty.	Check whether the API parameters in the code are correct.
4098	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.

4099	Recording is in progress.	Ensure that the SDK recording feature is used at the right time.
4100	Audio data is not captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	Mic permission is required for using the SDK. To add the permission, please see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
                switch (eventType) {
                    case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
                    {
                    // Process
                    break;
                        }
                    ...
                            case ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
                    {
                    // Process
                    break;
                    }
                }
}
```

Pausing recording

This API is used to pause recording. If you want to resume recording, please call the `ResumeRecording` API.

API prototype

```
ITMGPTT virtual int PauseRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->PauseRecording();
```

Resuming recording

This API is used to resume recording.

API prototype

```
ITMGPTT virtual int ResumeRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->ResumeRecording();
```

Canceling recording

This API is used to cancel recording. **There is no callback after cancellation**.

API prototype

```
ITMGPTT virtual int CancelRecording()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->CancelRecording();
```

Voice Message Upload, Download, and Playback

API	Description
UploadRecordedFile	Uploads the audio file
DownloadRecordedFile	Downloads the audio file
PlayRecordedFile	Plays back the audio file
StopPlayFile	Stops playing back the audio file
GetFileSize	Gets audio file size

GetVoiceFileDuration	Gets the audio file duration

Uploading an audio file

This API is used to upload an audio file.

API prototype

```
ITMGPTT virtual int UploadRecordedFile(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of uploaded audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->UploadRecordedFile(filePath);
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file
fileid	FString	File URL path

Error codes

Error Code Value	Cause	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.
8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
8195	A network error occurred.	Check whether the device can access the internet.

8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
        switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
            ...
                    case ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
            {
            // Process
            break;
            }
        }
}
```

Downloading the audio file

This API is used to download an audio file.

API prototype

```
ITMGPTT virtual int DownloadRecordedFile(const char* fileId, const char* filePath)
```

Parameter	Type	Description
fileId	const char*	URL path of file
filePath	const char*	Local path of saved file

Sample code

```
ITMGContextGetInstance()->GetPTT()->DownloadRecordedFile(fileID,filePath);
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	int32	0: recording is completed
filepath	FString	Path of stored recording file
fileid	FString	URL path of file, which will be retained on the server for 90 days

Error codes

Error Code Value	Cause	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
12291	Network storage system exception.	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	Server file system error.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed during the process of getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.

12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.
12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
        switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
```

Playing back audio

This API is used to play back audio.

API prototype

```
ITMGPTT virtual int PlayRecordedFile(const char* filePath)
ITMGPTT virtual int PlayRecordedFile(const char* filePath, nt voiceType)
```

Parameter	Type	Description
filePath	const char*	Local audio file path
voicetype	int	Voice changer type. For more information, see Voice Changing Effects.

Error codes

Error Code Value	Cause	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
ITMGContextGetInstance()->GetPTT()->PlayRecordedFile(filePath);
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameter includes `result` and `file_path` .

Parameter	Type	Description
code	int	0: playback is completed
filepath	FString	Path of stored recording file

Error codes

Error Code Value	Cause	Suggested Solution
20481	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Parameter is empty.	Check whether the API parameters in the code are correct.
20484	Internal error.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
    switch (eventType) {
    ...
        else if (eventType == ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE) {
            int32 result = JsonObject->GetIntegerField(TEXT("result"));
            FString filepath = JsonObject->GetStringField(TEXT("file_path"));
            onPttPlayFileCompleted(result, filepath);
        }
    }
}
```


Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

API prototype

```
ITMGPTT virtual int StopPlayFile()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->StopPlayFile();
```

Getting audio file size

This API is used to get the size of an audio file.

API prototype

```
ITMGPTT virtual int GetFileSize(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetFileSize(filePath);
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

API prototype

```
ITMGPTT virtual int GetVoiceFileDuration(const char* filePath)
```

Parameter	Type	Description
filePath	const char*	Path of audio file, which is a local path

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetVoiceFileDuration(filePath);
```

Fast Recording-to-Text Conversion

API	Description
SpeechToText	Converts speech to text

Converting audio file to text

This API is used to convert a specified audio file to text.

API prototype

```
ITMGPTT virtual void SpeechToText(const char* fileID)
```

Parameter	Type	Description
fileID	const char*	Audio file URL

Sample code

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(fileID);
```

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

API prototype

```
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage)
ITMGPTT virtual int SpeechToText(const char* fileID,const char* speechLanguage,cons
```

Parameter	Type	Description
fileID	const char*	The URL of the audio file, which will be retained on the server for 90 days.
speechLanguage	const char*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.

translatelanguage	const char*	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContextGetInstance()->GetPTT()->SpeechToText(filePath,"cmn-Hans-CN","cmn-Hans-C
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` and `text` (recognized text).

Parameter	Type	Description
result	int32	0: recording is completed
fileid	FString	URL of recording file, which will be retained on the server for 90 days
text	FString	Converted text

Error codes

Error Code Value	Cause	Suggested Solution
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.
32784	Incorrect speech-to-text conversion parameter.	Check whether the API parameter `fileid` in the code is empty.

32785	Speech-to-text translation returned an error.	Error with the backend of voice messaging and speech-to-text feature. Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

Sample code

```
void TMGTestScene::OnEvent(ITMG_MAIN_EVENT_TYPE eventType,const char* data){
        switch (eventType) {
            case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
            ...
                    case ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
            {
            // Process
            break;
            }
        }
}
```

Voice Message Volume Level APIs

API	Description
GetMicLevel	Gets real-time mic volume level
SetMicVolume	Sets recording volume level
GetMicVolume	Gets recording volume level

GetSpeakerLevel	Gets real-time speaker volume
SetSpeakerVolume	Sets playback volume level
GetSpeakerVolume	Gets playback volume level

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetMicLevel()
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetMicLevel();
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

API prototype

```
ITMGPTT virtual int SetMicVolume(int vol)
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->SetMicVolume(100);
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetMicVolume()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetMicVolume();
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetSpeakerLevel()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetSpeakerLevel();
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice message. Value range: 0-200.

API prototype

```
ITMGPTT virtual int SetSpeakerVolume(int vol)
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->SetSpeakerVolume(100);
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice message. An int-type value will be returned. Value range: 0-200.

API prototype

```
ITMGPTT virtual int GetSpeakerVolume()
```

Sample code

```
ITMGContextGetInstance()->GetPTT()->GetSpeakerVolume();
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for SDK usage analysis.

API prototype

```
ITMGContext virtual const char* GetSDKVersion()
```

Sample code

```
ITMGContextGetInstance()->GetSDKVersion();
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
ITMGContext int SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL levelPrint)
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

`ITMG_LOG_LEVEL` description:

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContextGetInstance()->SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

OS	Path
Windows	%appdata%\\Tencent\\GME\\ProcessName
iOS	Application/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/Documents
Android	/sdcard/Android/data/xxx.xxx.xxx/files
Mac	/Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents

API prototype

```
ITMGContext virtual int SetLogPath(const char* logDir)
```

Parameter	Type	Description
logDir	const char*	Path

Sample code

```
cosnt char* logDir = ""// Set a path by yourself
ITMGContext* context = ITMGContextGetInstance();
context->SetLogPath(logDir);
```

Callback Messages

Message	Description	Parameter
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was	result; error

	disconnected for network or other reasons	
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	The reconnection to the room started	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	The reconnection to the room succeeded	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status was changed	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker device was changed	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A new speaker device was added	result; error

ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker device was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A new mic device was added	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic device was lost	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic device was changed	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room quality changed	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Voice message recording was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Voice message upload was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Voice message download was completed	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Voice message playback was completed	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast speech-to-text conversion	result; text;file_id

	was completed	
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	Streaming speech-to-text conversion is in progress	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed	result; text;file_id

SDK for iOS

Integrating SDK

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to integrate GME SDK into an iOS project so that the iOS developers can easily debug and integrate the APIs for Game Multimedia Engine (GME).

SDK preparations

1. Download the applicable demo and SDK. For more information, see SDK Download Guide.
2. Decompress the obtained SDK resources.
3. The folder contains:

GMESDK.framework: Native resource related to iOS development.

libGMESDK.a: Unity resource related to iOS development (you can ignore this file if you don't use the Unity engine for development).

Note:

You can run the SDK on iOS 9.0 or later.

Voice Chat

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to integrate with and debug GME client APIs for the voice chat feature for iOS.

Key Considerations for Using GME

GME provides the real-time voice chat service, voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

Notes

You have created a GME application and obtained the `AppID` and `Key` of the SDK as instructed in Activating Services.

You have **activated the voice chat, voice messaging, and speech-to-text services of GME**. For more information, see Activating Services.

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being `0`.

GME APIs should be called in the same thread.

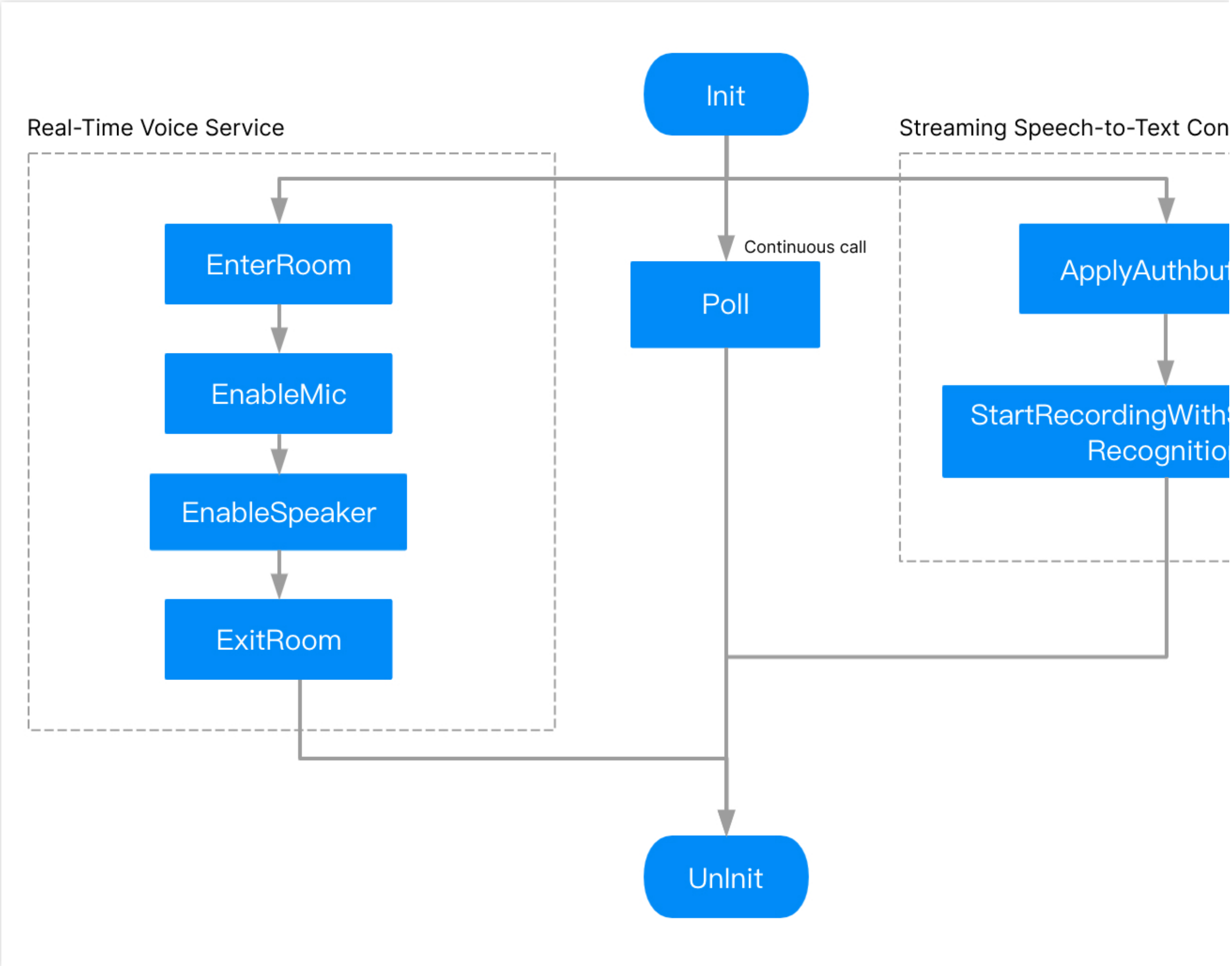
The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error codes, see Error Codes.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME
2. Calling Poll periodically to trigger callbacks
3. Entering a voice chat room
4. Turning on the mic
5. Turning on the speaker
6. Exiting the voice chat room
7. Uninitializing GME

APIs

```
@class ITMGRoom;//Room APIs
@class ITMGAudioCtrl;//Audio APIs
@class ITMGAudioEffectCtrl;//Sound effect and accompaniment APIs
```

Core APIs

API	Description
InitEngine	Initializes GME.
Poll	Triggers an event callback.
Pause	Pauses the system.
Resume	Resumes the system.
Uninit	Uninitializes GME.
SetDefaultAudienceAudioCategory	Sets the audio of the iOS device.

Note:

If you need to switch the account, call `UnInit` to uninitialize the SDK. No fees are incurred for calling the `InitEngine` API.

Imported header files

```
#import "GMESDK/TMGEngine.h"
#import "GMESDK/QAVAuthBuffer.h"
```

Getting singleton

To use the voice feature, get the `ITMGContext` object first.

```
+ (ITMGContext*) GetInstance;
```

Sample code

```
//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
```

Setting callbacks

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages.

Sample code

`ITMGDelegate` is used for declaration.

```
@interface TMGDemoViewController ()<ITMGDelegate>{}
```

```
ITMGDelegate < NSObject >

//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
_context.TMGDelegate = [DispatchCenter getInstance];
```

The API callback messages is processed in `OnEvent`. For the message type, see `ITMG_MAIN_EVENT_TYPE`. The message content is a dictionary for parsing the API callback contents.

API prototype

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data;
```

Sample code

```
//TMGRealTimeViewController.m
TMGRealTimeViewController ()< ITMGDelegate >


- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data {
    NSString *log = [NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventTyp
    [self showLog:log];
    NSLog(@"====%@====", log);
    switch (eventType) {
        // Step 6/11 : Perform the enter room event
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM: {
            int result = ((NSNumber *)[data objectForKey:@"result"]).intValue;
            NSString *error_info = [data objectForKey:@"error_info"];

            [self showLog:[NSString stringWithFormat:@"OnEnterRoomComplete:%d msg:(

            if (result == 0) {
                [self updateStatusEnterRoom:YES];
            }
        }
        break;
    }
}

// Refer to DispatchCenter.h and DispatchCenter.m
```

Initializing the SDK

You need to initialize the SDK through the `InitEngine` API before you can use the voice chat, voice messaging, and speech-to-text services. The `InitEngine` API must be called in the same thread as other APIs.

We recommend that you call all APIs in the main thread.

API prototype

```
-(int)InitEngine:(NSString*)sdkAppID openID:(NSString*)openID;
```

Parameter	Type	Description
sdkAppId	String	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
OpenId	String	`openID` can only be in Int64 type, which is passed in after being converted to a `const char*` . You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
QAV_OK= 0	The SDK was initialized successfully.
QAV_ERR_SDK_NOT_FULL_UPDATE= 7015	Checks whether the SDK file is complete. We recommend that you delete it and then import it again.

Notes on 7015 error code

The error code 7015 is identified based on the MD5 value. If this error is reported during integration, check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party enhancement, the Unity packaging mechanism, and other factors, the MD5 value of the library file will be affected, resulting in misjudgment. Therefore, **ignore this error in the logic for official releases**, and avoid displaying it on the UI.

Sample code

```
_openId = _userIdText.text;
_appId = _appIdText.text;
int result = 0;
// After the user consents to the application's privacy policy, initialize the SDK
//result = 0: The user consents to the application's privacy policy
//result = 1: The user does not consent to the application's privacy policy
// If the user does not consent to the privacy policy, change `ret` to a value othe
if (result == 0) {
```



```
    [[ITMGContext GetInstance] InitEngine:SDKAPPID openID:_openId];}
    else{
    log = [NSString stringWithFormat:@"The user does not consent to the application
}
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. `Poll` is the message pump of GME, and the `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.

Refer to the `EnginePollHelper.m` file in SDK Download Guide.

Call the `Poll` API periodically

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
-(void)Poll;
```

Sample code

```
[[ITMGContext GetInstance] Poll];
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
-(QAVResult)Pause;
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
-(QAVResult)Resume;
```

Uninitializing SDK

This API is used to uninitialize the SDK. **If the game account is bound to** `openid` , switching game account requires uninitializing GME and then using the new `openid` to initialize again.

Note:

If the end user revokes the permission granted to the SDK to process the personal information, you can call the `Uninit` API to stop using the SDK features and stop collecting and close the user data used by the features.

API prototype

```
-(int)Uninit;
```

Sample code

```
[[ITMGContext GetInstance] Uninit];
```

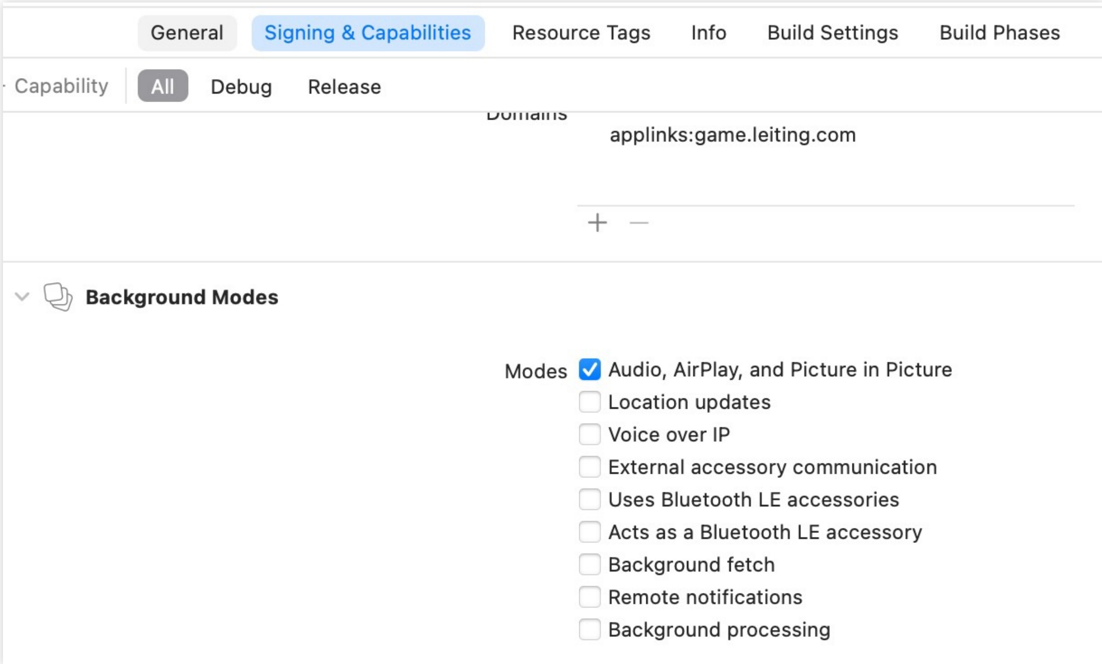
Audio settings for iOS device

This API is used to set the audio playback in the background, and the GME audio not to be affected by the mute switch or lock screen. For example, when the notification center or control center is opened, you can still receive and play back the GME audio. You need to call this API before room entry.

Meanwhile, you should pay attention to the following two points in the application:

Audio engine capture and playback are not paused when the application is switched to the background (i.e., `PauseAudio`).

You need to add at least `key:Required background modes` and `string:App plays audio or streams audio/video using AirPlay` to the `Info.plist` of the application.



Note:

It is recommended that developers call this API to set the audio.

Function prototype

```
-(QAVResult)SetDefaultAudienceAudioCategory:(ITMG_AUDIO_CATEGORY)audioCategory;
```

Type	Value	Description
ITMG_CATEGORY_AMBIENT	0	Audio is not played back in the background (default value).
ITMG_CATEGORY_PLAYBACK	1	Audio is played back in the background.

The specific implementation is to modify `kAudioSessionProperty_AudioCategory` . For more information, see Apple documentation.

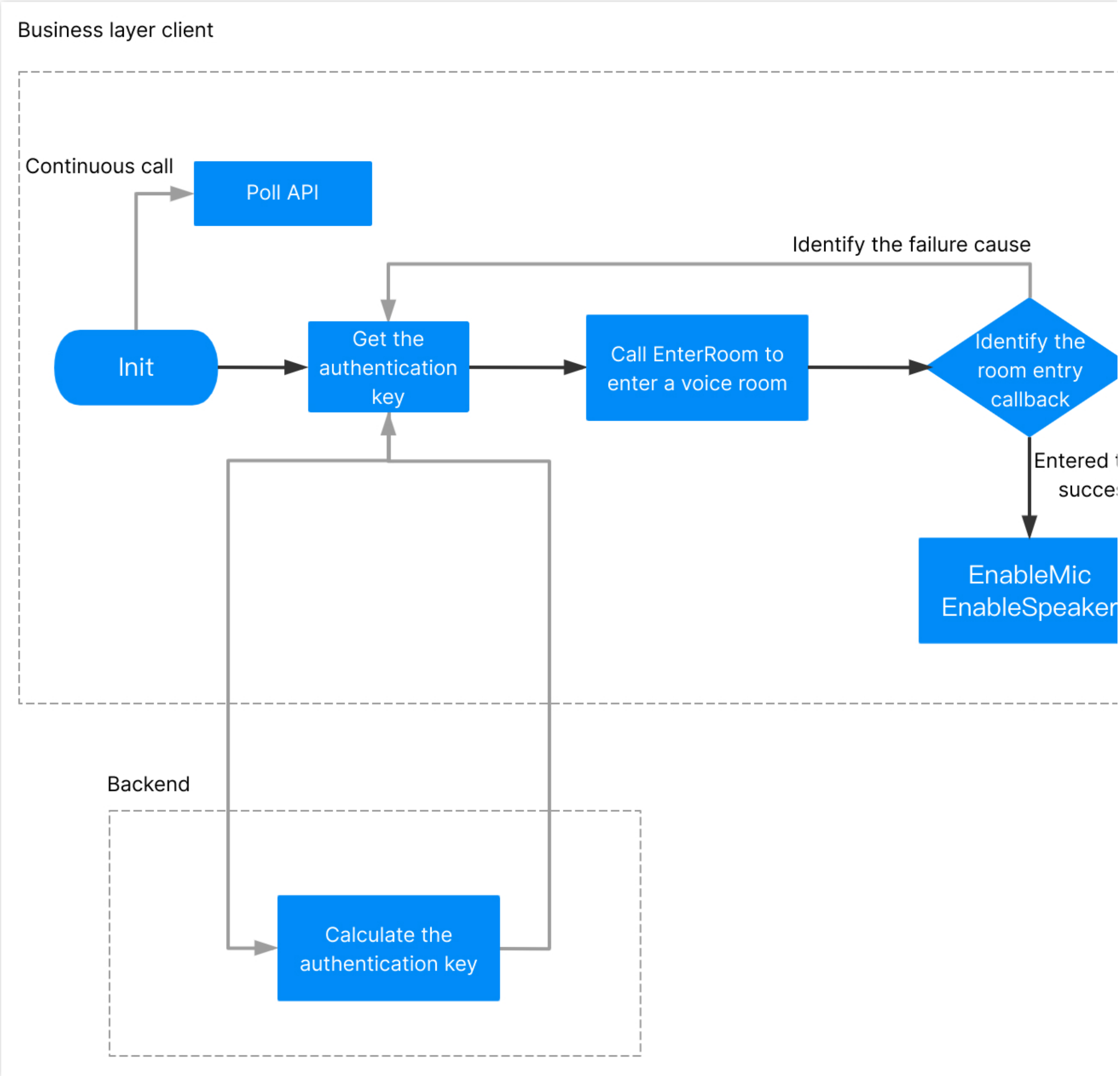
Sample code

```
[[ITMGContext GetInstance]SetDefaultAudienceAudioCategory:ITMG_CATEGORY_AMBIENT];
```

Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, see Sound and Audio.



API	Description
GenAuthBuffer	Calculates the local authentication key.
EnterRoom	Enters a room.
ExitRoom	Exits a room.
IsRoomEntered	Determines whether room entry is successful.
SwitchRoom	Switches the room quickly.
StartRoomSharing	Cross-room Co-anchoring

Local authentication key calculation

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.

API prototype

```
@interface QAVAuthBuffer : NSObject
+ (NSData*) GenAuthBuffer:(unsigned int)appId roomId:(NSString*)roomId openID:(NSSt
+ @end
```

Parameter	Type	Description
appId	unsigned int	`AppId` from the Tencent Cloud console
roomId	NSString *	Room ID, which can contain up to 127 characters.
openID	NSString *	User ID, which is the same as `openID` during initialization.
key	NSString *	Permission key from the Tencent Cloud console.

Sample code

```
#import "GMESDK/QAVAuthBuffer.h"
NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:SDKAPPID3RD.intValue roomId:_room
```

Entering a room

This API is used to enter a room with the generated authentication information. The mic and speaker are not enabled by default after room entry.

Note:

If the room entry callback result is `0`, the room entry is successful. If `0` is returned from the `EnterRoom` API, it doesn't necessarily mean that the room entry is successful.

The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will become smooth sound quality. Only when a member in the room calls the `ChangeRoomType` API, the audio type of the room will be changed.

Function prototype

```
-(int)EnterRoom:(NSString*) roomId roomType:(int)roomType authBuffer:(NSData*)authB
```

Parameter	Type	Description
roomId	NSString *	Room ID, which can contain up to 127 characters.
roomType	int	Room type. We recommend that you enter `ITMG_ROOM_TYPE_FLUENCY` . For more information on room audio types, see Sound Quality.
authBuffer	NSData *	Authentication key

Sample code

```
[[ITMGContext GetInstance] EnterRoom:_roomId roomType:_roomType authBuffer:authBuff
```

Callback for room entry

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function for callback and processing. A successful callback means that the room entry is successful, and the billing starts.

Billing references:

Purchase Guide

Billing

Will the billing continue if the client is disconnected from the server when using the voice chat?

Sample code

Sample code for processing the callback, including room entry and network disconnection events.

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
            int result = ((NSNumber*)[data objectForKey:@"result"]).intValue;
            NSString* error_info = [data objectForKey:@"error_info"];
                //Receive the event of successful room entry
        }
            break;
    }
}
```

Data details

Message	Data	Example

ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnected callback prompt `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` . At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` . When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` .

Error codes

Error Code	Cause and Suggested Solution
7006	Authentication failed. Causes: `AppID` doesn't exist or is incorrect. An error occurred while authenticating `authbuff` . Authentication expired. `OpenId` is invalid.
7007	The user was already in another room.
1001	The user was already in the process of entering a room but repeated this operation. We recommend that you not call the room entry API until the room entry callback is returned.
1003	The user was already in the room and called the room entry API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally.

Exiting a room

This API is used to exit the current room. It is an async API. The returned value `AV_OK` indicates a successful async delivery. If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API; instead, you can directly call the `EnterRoom` API.

API prototype

```
-(int)ExitRoom
```

Sample code

```
[[ITMGContext GetInstance] ExitRoom];
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM`.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
            // Receive the event of successful room exit
        }
            break;
    }
}
```

Data details

Message	Data	Example
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}

Determining whether user has entered room

This API is used to determine whether the user has entered a room. A value in bool type will be returned. Do not call this API during room entry.

API prototype

```
-(BOOL)IsRoomEntered;
```

Sample code

```
[[ITMGContext GetInstance] IsRoomEntered];
```

Switching room

User can call this API to quickly switch the voice chat room after entering the room. After the room is switched, the device is not reset, that is, if the microphone is already enabled in this room, the microphone will keep enabled after the room is switched.

The callback for quickly switching rooms is `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM`, and the fields are `error_info` and `result`.

API prototype

```
-(int) SwitchRoom:(NSString *)roomID  authBuffer:(NSData*)authBuffer;
```

Type descriptions

Parameter	Type	Description
targetRoomID	NSString *	ID of the room to enter
authBuffer	NSData*	Generates a new authentication key with the ID of the room to enter

Callback sample code

```
- (IBAction)swichRoom:(id)sender {
    NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:_appId.intValue roomID:_roomI
    [[[ITMGContext GetInstance]GetRoom]SwitchRoom:_roomIdText.text authBuffer:authB
}

-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSString* log =[NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventType
    [self showLog:log];
    NSLog(@"====%@====",log);
    switch (eventType) {
    case ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM:
        {
            int result = ((NSNumber*)[data objectForKey:@"result"]).intValue;
            NSString* log = nil;
            if (result == QAV_OK) {
                log = [NSString stringWithFormat:@"switch room success."];
            } else {
                log = [NSString stringWithFormat:@"switch room failed."];
            }
            [self showLog:log];
            break;
        }
    }
}
```

Cross-room mic connection

Call this API to connect the microphones across rooms after the room entry. After the call, the local user can communicate with the target OpenID user in the target room. The target room should be of the same type as the local room.

Example

User a is in room A, user b is in room B, and user a can talk with b through the cross-room API. When user c in room A speaks, users b and d in room B cannot hear. User c in room A can hear only the voice in room A and the voice of user b in room B but not other users in room B.

API prototype

```
-(int) StartRoomSharing:(NSString *)targetRoomID targetOpenID:(NSString *)targetOpe

-(int) StopRoomSharing;
```

Type descriptions

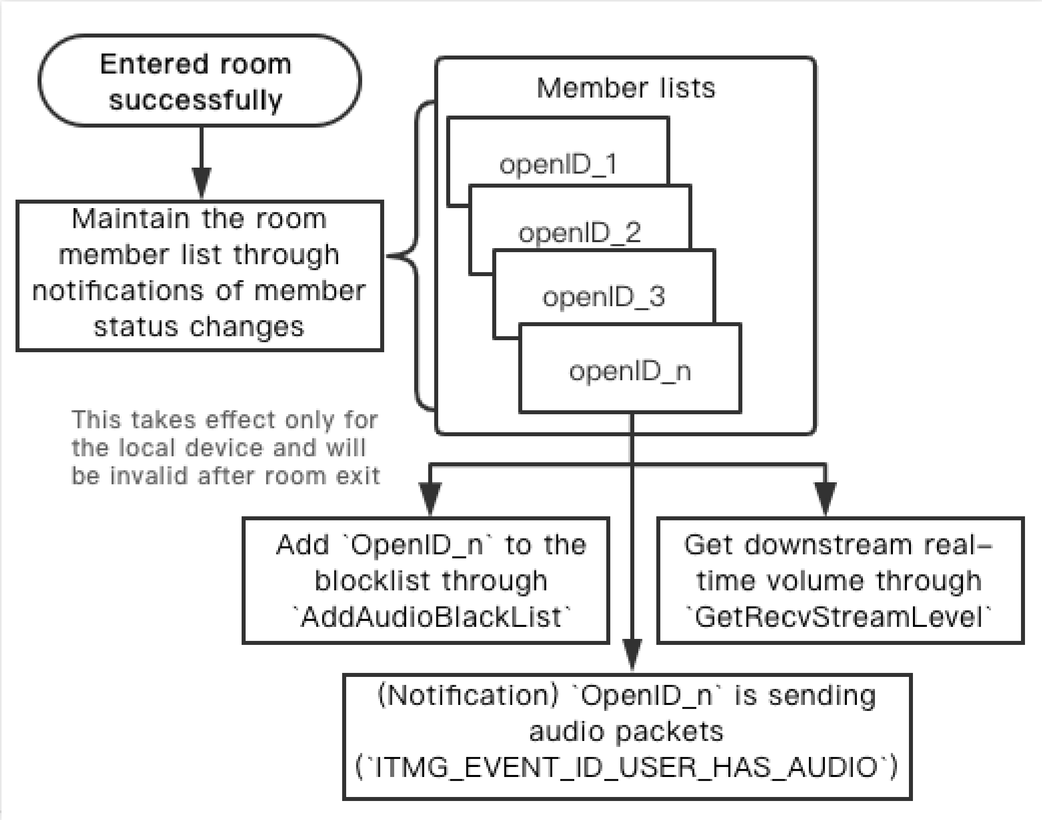
Parameter	Type	Description
targetRoomID	NSString *	ID of the room to connect mic
targetOpenID	NSString *	Target `OpenID` to connect mic
authBuffer	NSData*	Reserved flag. You just need to enter NULL.

Sample code

```
- (IBAction)shareRoom:(id)sender {
    if(_shareRoomSwitch.isOn){
        [[[ITMGContext GetInstance]GetRoom]StartRoomSharing:_shareRoomID.text targe
    }else{
        [[[ITMGContext GetInstance]GetRoom]StopRoomSharing];
    }
}
```

Room Status Maintenance

APIs in this section are used to display speaking members and members entering or exiting the room and mute a member in the room at the business layer.



API/Notification	Description
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The member status changed.
AddAudioBlackList	Mutes a member in the room.
RemoveAudioBlackList	Unmutes a member.

Notifications of member room entry and speaking status

This API is used to obtain the user speaking in the room and display it in the UI, and to send a notification when someone entering or exiting the room.

Notification for this event will be sent only when the status changes. To get member status in real time, cache the notification when it is received at a higher layer. The event message is `ITMG_MAIN_EVNET_TYPE_USER_UPDATE`, where the data contains `event_id` and `user_list`. The event message will be identified in the `OnEvent` function.

Notifications for audio events are subject to a threshold, and a notification will be sent only when this threshold is exceeded. The notification "A member has stopped sending audio packets" will be sent if no audio packets are received in more than two seconds.

event_id	Description	Maintenance
ITMG_EVENT_ID_USER_ENTER	Return the `openid` of the member entering the room.	Member list

ITMG_EVENT_ID_USER_EXIT	Return the `openid` of the member exiting the room.	Member list
ITMG_EVENT_ID_USER_HAS_AUDIO	Return the `openid` of the member sending audio packets in the room. This event can be used to determine whether a user is speaking and display the voiceprint effect.	Chat member list
ITMG_EVENT_ID_USER_NO_AUDIO	Return the `openid` of the member stopping sending audio packets in the room.	Chat member list

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    ITMG_EVENT_ID_USER_UPDATE event_id=((NSNumber*)[data objectForKey:@"event_id"])
    NSMutableArray* uses = [NSMutableArray arrayWithArray: [data objectForKey:@"use
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_USER_UPDATE:
        {
        // Process
        //Parse the parameter to get `event_id` and `user_list`
            switch (eventID)
             {
             case ITMG_EVENT_ID_USER_ENTER:
                  // A member enters the room
                  break;
             case ITMG_EVENT_ID_USER_EXIT:
                  // A member exits the room
                break;
            case ITMG_EVENT_ID_USER_HAS_AUDIO:
                // A member sends audio packets
                break;
            case ITMG_EVENT_ID_USER_NO_AUDIO:
                // A member stops sending audio packets
                break;
             }
        break;
        }
    }
}
```

Data details

Message	Data	Example

ITMG_MAIN_EVNET_TYPE_USER_UPDATE	event_id; user_list	{"event_id":0,"user_list":""}

Muting a member in the room

This API is used to add an ID to the audio data blocklist. This operation blocks audio from someone and only applies to the local device without affecting other devices. The returned value `0` indicates that the call is successful.

Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B.

If B blocks neither A nor C, B can hear both of them.

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

API prototype

```
ITMGContext GetAudioCtrl -(QAVResult)AddAudioBlackList:(NSString*)openID;
```

Parameter	Type	Description
openId	NSString	`openid` of the user to be blocked

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ] AddAudioBlackList[id]];
```

Unmuting

This API is used to remove an ID from the audio data blacklist. A returned value of 0 indicates the call is successful.

API prototype

```
-(QAVResult)RemoveAudioBlackList:(NSString*)openID;
```

Parameter	Type	Description
openId	NSString	User `openid` to be unblocked

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ] RemoveAudioBlackList[openId]];
```

Voice Chat Capturing APIs

The voice chat APIs can only be called after SDK initialization and room entry.

When the user clicks the button of enabling/disabling the mic or speaker on the UI, we recommend that you call the `EnableMic` or `EnableSpeaker` API.

When the user enters a voice chat room, enabling/disabling a capturing device will restart both capturing and playback devices. If the application is playing back background music, it will also be interrupted. Playback won't be interrupted if the mic is enabled/disabled through control of upstreaming/downstreaming. **Calling method: Call** `EnableAudioCaptureDevice(true)` and `EnableAudioPlayDevice(true)` once after room entry, and call `EnableAudioSend/Recv` to send/receive audio streams when Enable/Disable Mic is clicked.

API	Description
EnableMic	Enables/Disables the mic.
GetMicState	Gets the mic status.
EnableAudioCaptureDevice	Enables/Disables the capturing device.
IsAudioCaptureDeviceEnabled	Gets the capturing device status.
EnableAudioSend	Enables/Disables audio upstreaming.
IsAudioSendEnabled	Gets the audio upstreaming status.
GetMicLevel	Gets the real-time mic volume level.
GetSendStreamLevel	Gets the real-time audio upstreaming volume level.
SetMicVolume	Sets the mic volume level.
GetMicVolume	Gets the mic volume level.

Enabling or disabling mic

This API is used to enable/disable the mic. The mic and speaker are not enabled by default after room entry. `EnableMic` is equivalent to using `EnableAudioCaptureDevice` and `EnableAudioSend` together. **If accompaniment is used, call this API as instructed in** Accompaniment in Voice Chat.

API prototype

```
-(QAVResult)EnableMic:(BOOL)enable;
```

Parameter	Type	Description

enable	boolean	To enable the mic, set this parameter to `YES` . To disable the mic, set this parameter to `NO` .

Sample code

```
// Turn on mic
[[[ITMGContext GetInstance] GetAudioCtrl] EnableMic:YES];
```

Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

API prototype

```
-(int)GetMicState;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetMicState];
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.
This API can only be called after room entry. The device will be disabled automatically after room exit.
Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

API prototype

```
-(QAVResult)EnableAudioCaptureDevice:(BOOL)enabled;
```

Parameter	Type	Description
enabled	BOOL	To enable the capturing device, set this parameter to `YES` . To disable the capturing device, set this parameter to `NO` .

Sample code

```
// Enable capturing device
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioCaptureDevice:enabled];
```

Getting the capturing device status

This API is used to get the status of a capturing device.

API prototype

```
-(BOOL)IsAudioCaptureDeviceEnabled;
```

Sample code

```
BOOL IsAudioCaptureDevice = [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioCaptur
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain muted. For more information on how to enable/disable the capturing device, see the `EnableAudioCaptureDevice` API.

API prototype

```
-(QAVResult)EnableAudioSend:(BOOL)enable;
```

Parameter	Type	Description
enable	BOOL	To enable audio upstreaming, set this parameter to `YES` . To disable audio upstreaming, set this parameter to `NO` .

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioSend:enabled];
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

API prototype

```
-(BOOL)IsAudioSendEnabled;
```

Sample code

```
BOOL IsAudioSend = [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioSendEnabled];
```

Getting the real-time mic volume

This API is used to get the real-time mic volume. An int-type value in the range of 0-100 will be returned. It is recommended to call this API once every 20 ms.

API prototype

```
-(int)GetMicLevel;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetMicLevel];
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume. An int-type value in the range of 0-100 will be returned.

API prototype

```
-(int)GetSendStreamLevel();
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSendStreamLevel];
```

Setting the mic software volume

This API is used to set the mic volume level. The corresponding parameter is `volume` , which is equivalent to attenuating or gaining the captured sound.

API prototype

```
-(QAVResult)SetMicVolume:(int) volume;
```

Parameter	Type	Description
volume	int	Value range: 0-200. Default value: `100` . `0` indicates that the audio is muted, while `100` indicates that the volume level remains unchanged.

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] SetMicVolume:100];
```

Getting the mic software volume

This API is used to obtain the microphone volume. An "int" value is returned. Value 101 represents API SetMicVolume has not been called.

API prototype

```
-(int) GetMicVolume;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetMicVolume];
```

Voice Chat Playback APIs

API	Description
EnableSpeaker	Enables/Disables the speaker.
GetSpeakerState	Gets the speaker status.
EnableAudioPlayDevice	Enables/Disables the playback device.
IsAudioPlayDeviceEnabled	Gets the playback device status.
EnableAudioRecv	Enables/Disables audio downstreaming.
IsAudioRecvEnabled	Gets the audio downstreaming status.
GetSpeakerLevel	Gets the real-time speaker volume level.
GetRecvStreamLevel	Gets the real-time downstreaming audio levels of other members in the room.
SetSpeakerVolume	Sets the speaker volume level.
GetSpeakerVolume	Gets the speaker volume level.

Enabling or disabling speaker

This API is used to enable/disable the speaker. `EnableSpeaker` is equivalent to using `EnableAudioPlayDevice` and `EnableAudioRecv` together. **If accompaniment is used, call this API as instructed in** Accompaniment in Voice Chat.

API prototype

```
-(void)EnableSpeaker:(BOOL)enable;
```

Parameter	Type	Description
enable	boolean	To disable the speaker, set this parameter to `NO`. To enable the speaker, set this parameter to `YES`.

Sample code

```
// Turn on the speaker
[[[ITMGContext GetInstance] GetAudioCtrl] EnableSpeaker:YES];
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, and 1 is on.

Function prototype

```
-(int)GetSpeakerState;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSpeakerState];
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

API prototype

```
-(QAVResult)EnableAudioPlayDevice:(BOOL)enabled;
```

Parameter	Type	Description
enabled	BOOL	To disable the playback device, set this parameter to `NO`. To enable the playback device, set this parameter to `YES`.

Sample code

```
// Enable the playback device
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioPlayDevice:enabled];
```

Getting the playback device status

This API is used to get the status of a playback device.

API prototype

```
-(BOOL)IsAudioPlayDeviceEnabled;
```

Sample code

```
BOOL IsAudioPlayDevice =  [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioPlayDevi
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain muted. For more information on how to enable/disable the playback device, see the `EnableAudioPlayDevice` API.

API prototype

```
-(QAVResult)EnableAudioRecv:(BOOL)enabled;
```

Parameter	Type	Description
enabled	BOOL	To enable audio downstreaming, set this parameter to `YES`. To disable audio downstreaming, set this parameter to `NO`.

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioRecv:enabled];
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

API prototype

```
-(BOOL)IsAudioRecvEnabled;
```

Sample code

```
BOOL IsAudioRecv = [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioRecvEnabled];
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume. An int-type value will be returned to indicate the volume. It is recommended to call this API once every 20 ms.

API prototype

```
-(int)GetSpeakerLevel;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSpeakerLevel];
```

Getting the real-time downstreaming audio levels of other members in room

This API is used to get the real-time audio downstreaming volume of other members in the room. An int-type value will be returned. Value range: 0-200.

API prototype

```
-(int)GetRecvStreamLevel:(NSString*) openID;
```

Parameter	Type	Description
openID	NSString	`openId` of another member in the room

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetRecvStreamLevel:(NSString*) openId];
```

Dynamically setting the volume of a member of the room

This API is used to set the volume of a member in the room. It takes effect only on the local.

API prototype

```
-(int) SetSpeakerVolumeByOpenID:(NSString *)openId volume:(int)volume;
```

Parameter	Type	Description
openId	String *	`OpenID` of the target user
volume	int	Percentage. Recommended value range: 0-200. Default value: `100` .

Getting volume percentage

Call this API to get the volume set by SetSpeakerVolumeByOpenID

API prototype

```
-(int) GetSpeakerVolumeByOpenID:(NSString *)openId;
```

Returned values

API returns volume percentage set by OpenID, where 100 is by default.

Setting the speaker volume

This API is used to set the speaker volume.

The corresponding parameter is volume. 0 indicates that the audio is muted, while 100 indicates that the volume remains unchanged. The default value is 100.

Function prototype

```
-(QAVResult)SetSpeakerVolume:(int)vol;
```

Parameter	Type	Description
vol	int	Volume level. Value range: 0–200.

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] SetSpeakerVolume:100];
```

Getting the speaker volume

This API is used to get the speaker volume. An int-type value will be returned to indicate the volume. 101 indicates that the `SetSpeakerVolume` API has not been called.

"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

Function prototype

```
-(int)GetSpeakerVolume;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSpeakerVolume];
```

Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

Function prototype

```
-(QAVResult)EnableLoopBack:(BOOL)enable;
```

Parameter	Type	Description
enable	boolean	Specifies whether to enable the in-ear monitoring.

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] EnableLoopBack:YES];
```

Modifying user's room audio type

This API is used to modify a user's room audio type. For the result, please see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

Function prototype

```
-(int)ChangeRoomType:(int)nRoomType;
```

Parameter	Type	Description
nRoomType	int	Room type to be switched to. For room audio types, see the `EnterRoom` API.

Sample code

```
[[[ITMGContext GetInstance]GetRoom ]ChangeRoomType:_roomType];
```

Getting user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, please see the `EnterRoom` API.

Function prototype

```
-(int)GetRoomType;
```

Sample code

```
[[[ITMGContext GetInstance]GetRoom ]GetRoomType];
```

Callback for modifying the room type

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result`, `error_info`, and `new_room_type`. The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	The existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	A user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	A user is already in the room, and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	A room member calls the `ChangeRoomType` API to request a change of the room audio type.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
```

```
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE:
            NSLog(@"ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE:%@ ",data);
            int result = ((NSNumber*)[data objectForKey:@"result"]).intValue;
            int newRoomType = ((NSNumber*) [data objectForKey:@"new_room_type"]).in
            int subEventType = ((NSNumber*) [data objectForKey:@"sub_event_type"]).
     }
}
```

Data details

Message	Data	Exa
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result;error_info;new_room_type;subEventType	{"e

The monitoring event of room call quality

The message for quality monitoring event triggered once every two seconds after room entry is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY` . The returned parameters include `weight` , `loss` , and `delay` , which represent the following information.

This API is used to monitor the network quality. If the user's network is poor, the business layer will remind the user to switch to a better network through the UI.

Parameter	Type	Description
weight	int	Value range: 1–50. `50` indicates excellent sound quality, `1` indicates very poor (barely usable) sound quality, and `0` represents an initial meaningless value. Generally, if the value is below 30, you can remind users that the network is poor and recommend them to switch the network.
Loss	double	Upstream packet loss rate
Delay	int	Voice chat delay in ms

Getting version number

This API is used to get the SDK version number for analysis.

Function prototype

```
-(NSString*)GetSDKVersion;
```

Sample code


```
[[ITMGContext GetInstance] GetSDKVersion];
```

Checking mic permission

This API is used to return the mic permission status.

Function prototype

```
-(ITMG_RECORD_PERMISSION)CheckMicPermission;
```

Parameter description

Parameter	Value	Description
ITMG_PERMISSION_GRANTED	0	The mic permission is granted.
ITMG_PERMISSION_Denied	1	Microphone disabled.
ITMG_PERMISSION_NotDetermined	2	No authorization box has been popped up to request the permission.
ITMG_PERMISSION_ERROR	3	An error occurred while calling the API.

Sample code

```
[[ITMGContext GetInstance] CheckMicPermission];
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

Function prototype

```
-(void)SetLogLevel:(ITMG_LOG_LEVEL)levelWrite (ITMG_LOG_LEVEL)levelPrint;
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO` .
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE`

		indicates not to print. Default value: `TMG_LOG_LEVEL_ERROR` .

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
[[ITMGContext GetInstance] SetLogLevel:TMG_LOG_LEVEL_INFO TMG_LOG_LEVEL_INFO];
```

Setting the log printing path

This API is used to set the log printing path, and needs to be called before initialization. The default path is `Application/********-****-****-****-************/Documents` .

Function prototype

```
-(void)SetLogPath:(NSString*)logDir;
```

Parameter	Type	Description
logDir	NSString	Path

Sample code

```
[[ITMGContext GetInstance] SetLogPath:Path];
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

Function prototype

```
-(NSString*)GetQualityTips;
```

Sample code

```
[[[ITMGContext GetInstance]GetRoom ] GetQualityTips];
```

Callback Messages

Message list

Message	Description
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room.
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room.
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected due to a network or another issue.
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room type changed.
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated.
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room quality changed.

Data list

Message	Data	Example
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"",
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"",
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"w
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; sub_event_type; new_room_type	{"error_info":"",
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	result; error_info	{"deviceID":"{0 dd00542b47ae Audio)","error_
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	result; error_info	{"deviceID":"{0 dd00542b47ae

		Audio)","error_
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	result; error_info	{"deviceID":"{0 7e454093f229 Audio)","error_
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	result; error_info	{"deviceID":"{0 7e454093f229 Audio)","error_
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	user_list; event_id	{"event_id":1,"u
ITMG_MAIN_EVENT_TYPE_NUMBER_OF_USERS_UPDATE	AllUser; AccUser; ProxyUser	{"AllUser":3,"A
ITMG_MAIN_EVENT_TYPE_NUMBER_OF_AUDIOSTREAMS_UPDATE	AudioStreams	{"AudioStream
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	weight; loss; delay	{"weight":5,"los

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:11:45

This document describes how to integrate with and debug the GME APIs for iOS.

Note:

This document applies to GME SDK version 2.9.

Key Considerations for Using GME

GME provides two services: Voice chat service and voice messaging and speech-to-text service, both of which rely on key APIs such as Init and Poll.

Note:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.

Non-streaming speech-to-text API ***SpeechToText()***: There can be up to 10 concurrent requests per account.

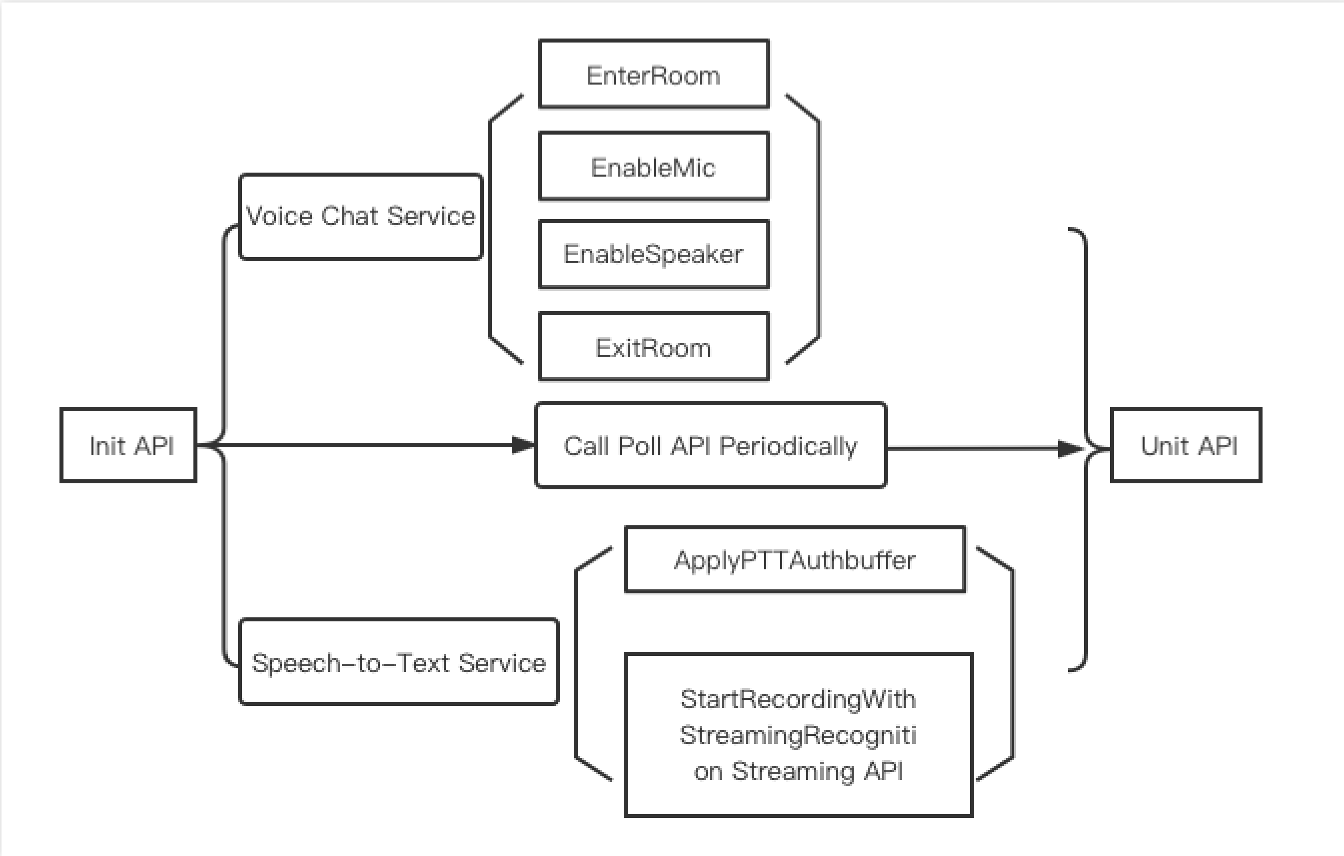
Streaming speech-to-text API ***StartRecordingWithStreamingRecognition()***: There can be up to 50 concurrent requests per account.

Real-time streaming speech-to-text API ***StartRealTimeASR()***: There can be up to 50 concurrent requests per account.

Note on Init API

If you need to use voice chat and voice message services at the same time, **you only need to call `Init` API once**.

The billing will not start after initialization. Receiving or sending a voice message in speech-to-text service is counted as a voice message DAU.



Directions

1. Initializing GMEAPI: Init
2. Calling Poll periodically to trigger event callbacksAPI: Poll
3. Initializing authenticationAPI: ApplyPTTAuthbuffer
4. Starting streaming speech recognitionAPI: StartRecordingWithStreamingRecognition
5. Stop recordingAPI: StopRecording
6. Uninitializing GMEAPI: UnInit

Important notes

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error code, see Error Codes.

Core APIs

Before the initialization, the SDK is in the uninitialized status, and **you need to initialize it through the `Init` API** before you can use the voice chat and speech-to-text services.

Call the `Init` API before calling any APIs of GME.

If you have any questions when using the service, see General.

API	Description
InitEngine	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME
SetDefaultAudienceAudioCategory	Sets audio playback in background on device

Imported header files

```
#import "GMESDK/TMGEngine.h"
#import "GMESDK/QAVAuthBuffer.h"
```

Getting singleton

To use the voice feature, get the `ITMGContext` object first.

```
+ (ITMGContext*) GetInstance;
```

Sample code

```
//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
```

Setting callbacks

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages.

Sample code

`ITMGDelegate` is used for declaration.

```
@interface TMGDemoViewController ()<ITMGDelegate>{}
```

```
ITMGDelegate < NSObject >

//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
_context.TMGDelegate = [DispatchCenter getInstance];
```

The API callback messages is processed in `OnEvent`. For the message type, see `ITMG_MAIN_EVENT_TYPE`. The message content is a dictionary for parsing the API callback contents.

Function prototype

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data;
```

Sample code

```
//TMGRealTimeViewController.m
TMGRealTimeViewController ()< ITMGDelegate >


- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data {
    NSString *log = [NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventTyp
    [self showLog:log];
    NSLog(@"====%@====", log);
    switch (eventType) {
        // Step 6/11 : Perform the enter room event
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM: {
            int result = ((NSNumber *)[data objectForKey:@"result"]).intValue;
            NSString *error_info = [data objectForKey:@"error_info"];

            [self showLog:[NSString stringWithFormat:@"OnEnterRoomComplete:%d msg:(

            if (result == 0) {
                [self updateStatusEnterRoom:YES];
            }
        }
        break;
    }
}

// Refer to DispatchCenter.h and DispatchCenter.m
```

Initializing SDK

This API is used to initialize the GME service. It is recommended to call it when initializing the application. No fee is incurred for calling this API.

For more information on how to get the `sdkAppID` parameter, see Activating Services.
The openID uniquely identifies a user with the rules stipulated by the application developer and unique in the application (currently, only INT64 is supported).

Note:

The Init API must be called in the same thread with other APIs. It is recommended to call all APIs in the main thread.

Function prototype

```
-(int)InitEngine:(NSString*)sdkAppID openID:(NSString*)openID;
```

Parameter	Type	Description
sdkAppId	String	`AppId` provided by the GME service from the Tencent Cloud console
OpenId	String	`OpenId` can only be in Int64 type, which is passed after being converted to a string.

Returned Value	Description
QAV_OK= 0	Initialized SDK successfully.
QAV_ERR_SDK_NOT_FULL_UPDATE= 7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is only a reminder but will not cause an initialization failure.
If this error is reported during integration, please check the integrity and version of the SDK file as prompted.
If this error is returned after executable file export, please ignore it and try to avoid displaying it in the UI.

Sample code

```
_openId = _userIdText.text;
_appId = _appIdText.text;
[[ITMGContext GetInstance] InitEngine:SDKAPPID openID:_openId];
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. The `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.
Refer to the EnginePollHelper.m file in Demo.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

Function prototype


```
-(void)Poll;
```

Sample code

```
[[ITMGContext GetInstance] Poll];
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause.
If you need to pause the audio when switching to the background, you can call the `Pause` API in the listening code used to switch to the background, and call the `Resume` API in the listening event used to resume the foreground.

Function prototype

```
-(QAVResult)Pause;
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

Function prototype

```
-(QAVResult)Resume;
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **Switching accounts requires uninitialization**.

Function prototype

```
-(int)Uninit;
```

Sample code

```
[[ITMGContext GetInstance] Uninit];
```

Audio settings for iOS device

This API is used to set the audio playback in the background, and the GME audio not to be affected by the mute switch or lock screen. For example, when the notification center or control center is opened, you can still receive and play back the GME audio. You need to call this API before room entry.

Meanwhile, you should pay attention to the following two points in the application:

Audio engine capture and playback are not paused when the application is switched to the background (i.e., `PauseAudio`).

You need to add at least `key:Required background modes` and `string:App plays audio or streams audio/video using AirPlay` to the `Info.plist` of the application.

Note:

It is recommended that developers call this API to set the audio.

Function prototype

```
-(QAVResult)SetDefaultAudienceAudioCategory:(ITMG_AUDIO_CATEGORY)audioCategory;
```

Type	Parameter	Description
ITMG_CATEGORY_AMBIENT	0	Audio is not played back in the background (default value)
ITMG_CATEGORY_PLAYBACK	1	Audio is played back in the background

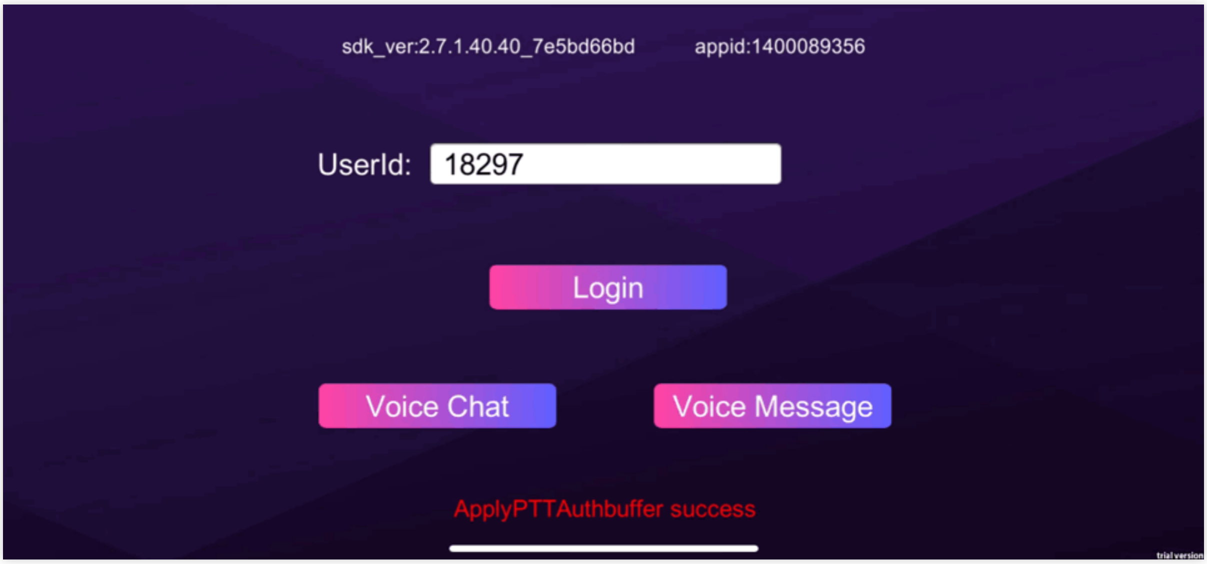
This can be achieved by modifying kAudioSessionProperty_AudioCategory. For more information, see Apple official documentation.

Sample code

```
[[ITMGContext GetInstance]SetDefaultAudienceAudioCategory:ITMG_CATEGORY_AMBIENT];
```

Speech-to-Text

Voice messaging refers to recording and sending a voice message. At the same time, the voice message can be converted to text and translated, as shown below:

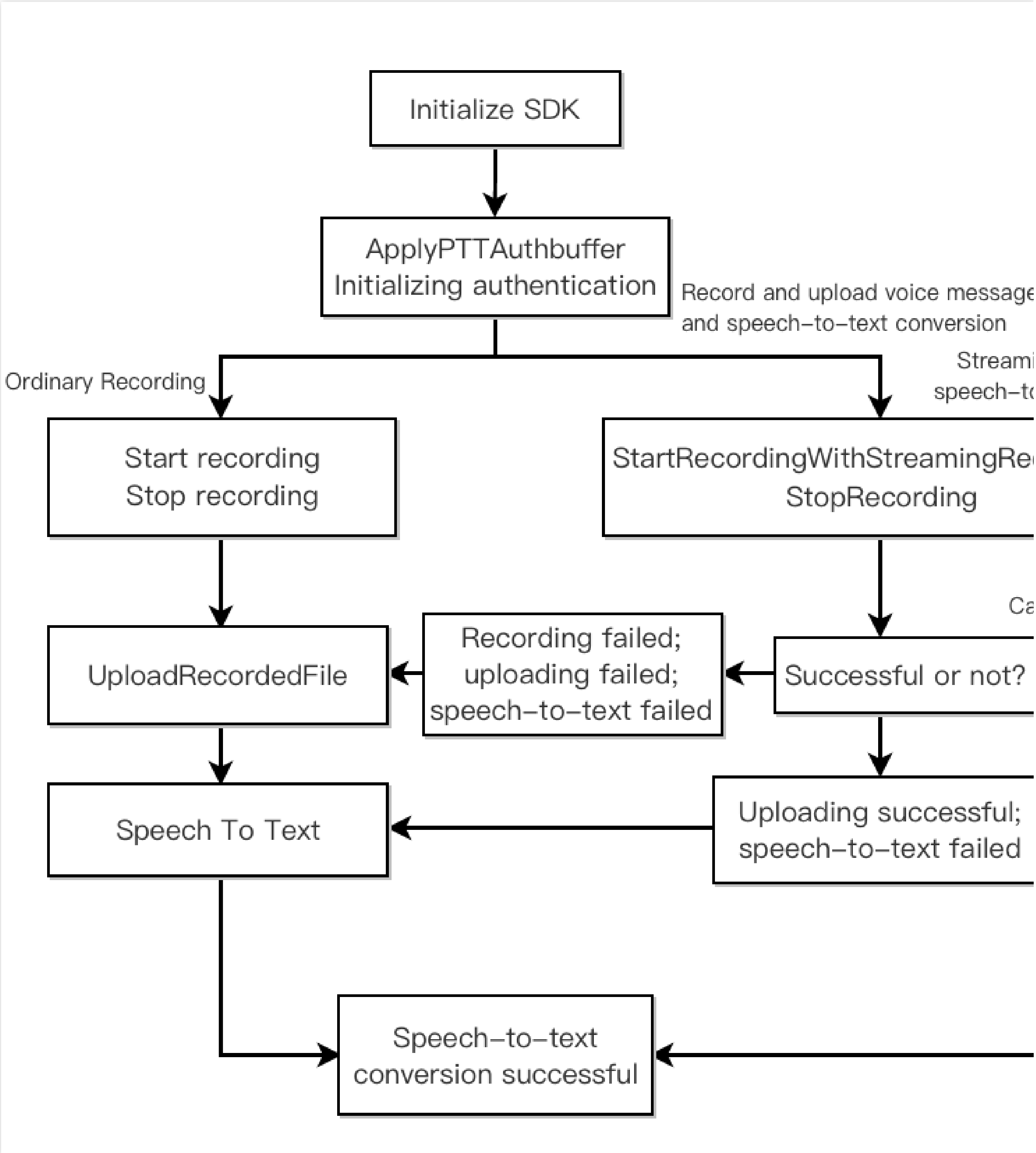


Note:

It is recommended to use the streaming speech-to-text service.

You do not need to enter a voice chat room when using the voice messaging service.

Voice message and speech-to-text conversion flowchart



Integrating Voice Messaging and Speech-to-Text Service

Voice messaging and speech-to-text APIs

API	Description

ApplyPTTAuthbuffer	Initializes authentication
SetMaxMessageLength	Specifies the maximum length of voice message
StartRecording	Starts recording
StartRecordingWithStreamingRecognition	Starts streaming recording
PauseRecording	Pauses recording
ResumeRecording	Resumes recording
StopRecording	Stops recording
CancelRecording	Cancels recording
GetMicLevel	Gets the real-time mic volume
SetMicVolume	Sets the recording volume
GetMicVolume	Gets the recording volume
GetSpeakerLevel	Gets the real-time speaker volume
SetSpeakerVolume	Sets the playback volume
GetSpeakerVolume	Gets the playback volume
UploadRecordedFile	Uploads the audio file
DownloadRecordedFile	Downloads the audio file
PlayRecordedFile	Plays back audio
StopPlayFile	Stops playing back audio
GetFileSize	Gets the audio file size
GetVoiceFileDuration	Gets the audio file duration
SpeechToText	Converts speech to text

Maximum recording duration

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, please call the `SetMaxMessageLength` API to set it after initialization.

Initializing the SDK

Before the initialization, the SDK is in the uninitialized status, and you need to initialize it through the `Init` API before you can use the voice chat and voice message services.
If you have any questions when using the service, see Speech-to-text Conversion.

Authentication information

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.
To get authentication for voice message and speech-to-text, the room ID parameter must be set to `null` .

Function prototype

```
@interface QAVAuthBuffer : NSObject
+ (NSData*) GenAuthBuffer:(unsigned int)appId roomId:(NSString*)roomId openID:(NSSt
+ @end
```

Parameter	Type	Description
appId	int	`AppId` from the Tencent Cloud console.
roomId	NSString	Enter `null` .
openID	NSString	User ID, which is the same as `openID` during initialization.
key	NSString	Permission key from the Tencent Cloud console.

Sample code

```
#import "GMESDK/QAVAuthBuffer.h"
NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:SDKAPPID3RD.intValue roomId:_room
```

Initializing authentication

Call authentication initialization after initializing the SDK. For more information on how to get the `authBuffer` , please see `genAuthBuffer` (the voice chat authentication information API).

Function prototype

```
public abstract int ApplyPTTAuthbuffer(byte[] authBuffer);
```

Parameter	Type	Description

authBuffer	NSData*	Authentication

Sample code

```
[[[ITMGContext GetInstance]GetPTT]ApplyPTTAuthbuffer:(NSData *)authBuffer];
```

Streaming Speech Recognition

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call** `StopRecording` . The callback will be returned after the recording is stopped.

Function prototype

```
-(int)StartRecordingWithStreamingRecognition:(NSString *)filePath;
-(int)StartRecordingWithStreamingRecognition:(NSString *)filePath language:(NSStrin
```

Parameter	Type	Description
filePath	String	Path of stored audio file
speechLanguage	String	The language in which the audio file is to be converted to text. For parameters, please see Language Parameter Reference List
translateLanguage	String	The language into which the audio file will be translated. For parameters, please see Language Parameter Reference List (This parameter is currently unavailable. Enter the same value as that of `speechLanguage`)

Sample code

```
recordfilePath = [docDir stringByAppendingFormat:@"/test_%d.ptt",index++];
[[[ITMGContext GetInstance] GetPTT] StartRecordingWithStreamingRecognition:recordfi
```

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen for callback messages in the callback function `onEvent`. Event messages are divided into:
`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.
`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real-time during the recording, which is equivalent to returning the recognized text while speaking.
The event message will be identified in the `OnEvent function` based on the actual needs. The passed parameters include the following four messages.

Message Name	Description
result	A return code for judging whether the streaming speech recognition is successful.
text	Text converted from speech
file_path	Local path of stored recording file
file_id	Backend URL address of recording file, which will be retained for 90 days. `fileid` is fixed at `http://gme-v2-`

Note:
The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Suggested Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.

Sample code

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data
```

```
{
    NSNumber *number = [data objectForKey:@"result"];
    switch (eventType)
    {
        case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                donwLoadUrlPath = data[@"file_id"];

                recordfilePath = [data objectForKey:@"file_path"];
                _localFileField.text = recordfilePath;

                _donwloadUrlField.text = [data objectForKey:@"file_id"] ;

                UITextField *_audiotoTextField =(UITextField*)objc_getAssociatedObj
                _audiotoTextField.text = [data objectForKey:@"text"] ;
            }

        }
            break;
    }
}
```

Voice Message Recording

The recording process is as follows: start recording -> stop recording -> return recording callback -> start the next recording.

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

Function prototype

```
-(QAVResult)SetMaxMessageLength:(int)msTime
```

Parameter	Type	Description
msTime	int	Audio duration in ms. Value range: 1000 < msTime <= 58000

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SetMaxMessageLength:(int)msTime];
```

Starting recording

This API is used to start recording. The recording file must be uploaded first before you can perform operations such as speech-to-text conversion. **To stop recording, call `StopRecording`**.

Function prototype

```
-(int)StartRecording:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of stored audio file

Sample code

```
recordfilePath =[docDir stringByAppendingFormat:@"/test_%d.ptt",index++];
[[[ITMGContext GetInstance]GetPTT]StartRecording:recordfilePath];
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

Function prototype

```
-(QAVResult)StopRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]StopRecording];
```

Callback for recording start

A callback will be executed through a delegate function to pass a message when recording is completed.

To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.
The callback function `OnEvent` will be called after recording is started. The event message `ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameter includes `result` and `file_path` .

Error codes

Error Code Value	Cause	Suggested Solution
4097	Parameter is empty.	Check whether the API parameters in the code are correct.
4098	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
4099	Recording is in progress.	Ensure that the SDK recording feature is used at the right time.
4100	Audio data is not captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	Mic permission is required for using the SDK. To add the permission, please see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
        {
        //Recording callback
```



```
        }
            break;
    }
}
```

Pausing recording

This API is used to pause recording. If you want to resume recording, please call the `ResumeRecording` API.

Function prototype

```
-(int)PauseRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]PauseRecording];
```

Resuming recording

This API is used to resume recording.

Function prototype

```
-(int)ResumeRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]ResumeRecording];
```

Canceling recording

This API is used to cancel recording. There is no callback after cancellation.

Function prototype

```
-(QAVResult)CancelRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]CancelRecording];
```

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(QAVResult)GetMicLevel;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetMicLevel];
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(QAVResult)SetMicVolume:(int) volume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SetMicVolume:100];
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(int)GetMicVolume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetMicVolume];
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(QAVResult)GetSpeakerLevel;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetSpeakerLevel];
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice messaging. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(QAVResult)SetSpeakerVolume:(int)volume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SetSpeakerVolume:100];
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice messaging. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(int)GetSpeakerVolume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetSpeakerVolume];
```

Voice Message Playback

Playing back audio

This API is used to play back audio.

Function prototype

```
-(int)PlayRecordedFile:(NSString*)filePath;
-(int)PlayRecordedFile:(NSString*)filePath VoiceType:(ITMG_VOICE_TYPE) type;
```

Parameter	Type	Description
downloadFilePath	NSString	Local audio file path
type	ITMG_VOICE_TYPE	Voice changer type. For more information, see Voice Changing Effects.

Error codes

Error Code Value	Cause	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]PlayRecordedFile:path];
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameter includes `result` and `file_path` .

Error codes

Error Code Value	Cause	Suggested Solution
20481	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Parameter is empty.	Check whether the API parameters in the code are correct.
20484	Internal error.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE:
        {
        // Callback for audio playback
        }
            break;
    }
}
```

Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

Function prototype

```
-(int)StopPlayFile;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]StopPlayFile];
```

Getting audio file size

This API is used to get the size of an audio file.

Function prototype

```
-(int)GetFileSize:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of audio file, which is a local path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetFileSize:path];
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

Function prototype

```
-(int)GetVoiceFileDuration:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of audio file, which is a local path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetVoiceFileDuration:path];
```

Voice Message Upload and Download

Uploading an audio file

This API is used to upload an audio file.

Function prototype

```
-(void)UploadRecordedFile:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of uploaded audio file, which is a local path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]UploadRecordedFile:path];
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameters include `result` , `file_path` , and `file_id` .

Error codes

Error Code Value	Cause	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.

8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
8195	A network error occurred.	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                _donwloadUrlField.text = [data objectForKey:@"file_id"] ;
                donwLoadUrlPath = [data objectForKey:@"file_id"] ;
            }
        }
            break;
    }
}
```

Downloading the audio file

This API is used to download an audio file.

Function prototype

```
-(void)DownloadRecordedFile:(NSString*)fileId downloadFilePath:(NSString*)downloadF
```

Parameter	Type	Description
fileID	NSString	File URL path
downloadFilePath	NSString	Local path of saved file

Sample code

```
[[[ITMGContext GetInstance]GetPTT]DownloadRecordedFile:fileIdpath downloadFilePath:
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Error codes

Error Code Value	Cause	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
12291	Network storage system exception	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	Server file system error.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed during the process of getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.

12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.
12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                _audiofileToPlayField.text = [data objectForKey:@"file_path"] ;
                donwLoadLocalPath = [data objectForKey:@"file_path"];
            }
            else
            {
                donwLoadLocalPath = NULL;
            }
        }
            break;
    }
}
```

Speech-to-Text Service

Converting audio file to text

This API is used to convert a specified audio file to text.

Function prototype

```
-(void)SpeechToText:(NSString*)fileID;
```

Parameter	Type	Description
fileID	NSString	URL of audio file

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SpeechToText:fileID];
```

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Function prototype

```
-(void)SpeechToText:(NSString*)fileID (NSString*)speechLanguage (NSString*)translat
```

Parameter	Type	Description
fileID	NSString*	URL of audio file, which will be retained on the server for 90 days
speechLanguage	NSString*	The language in which the audio file is to be converted to text. For parameters, please see Language Parameter Reference List.
translateLanguage	NSString*	The language into which the audio file will be translated. For parameters, please see Language Parameter Reference List. This parameter is currently unavailable. Enter the same value as that of `speechLanguage` .

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SpeechToText:fileID speechLanguage:"cmn-Hans-CN"
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` and `text` (recognized text).

Error codes

Error Code	Cause	Suggested Solution

Value		
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.
32784	Incorrect speech-to-text conversion parameter.	Check whether the API parameter `fileid` in the code is empty.
32785	Speech-to-text translation returned an error.	Error with the backend of voice message and speech-to-text feature. Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                UITextField *_audiotoTextField =(UITextField*)objc_getAssociatedObj
                _audiotoTextField.text = [data objectForKey:@"text"] ;
            }
        }
            break;
    }
}
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for analysis.

Function prototype

```
-(NSString*)GetSDKVersion;
```

Sample code

```
[[ITMGContext GetInstance] GetSDKVersion];
```

Checking mic permission

This API is used to return the mic permission status.

Function prototype

```
-(ITMG_RECORD_PERMISSION)CheckMicPermission;
```

Parameter description

Parameter	Value	Description
ITMG_PERMISSION_GRANTED	0	Mic permission is granted.
ITMG_PERMISSION_Denied	1	Mic is disabled.
ITMG_PERMISSION_NotDetermined	2	No authorization box has been popped up to request the permission.
ITMG_PERMISSION_ERROR	3	An error occurred while calling the API.

Sample code

```
[[ITMGContext GetInstance] CheckMicPermission];
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

Function prototype

```
-(void)SetLogLevel:(ITMG_LOG_LEVEL)levelWrite (ITMG_LOG_LEVEL)levelPrint;
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

ITMG_LOG_LEVEL

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
[[ITMGContext GetInstance] SetLogLevel:TMG_LOG_LEVEL_INFO TMG_LOG_LEVEL_INFO];
```

Setting the log printing path

This API is used to set the log printing path, and needs to be called before initialization. The default path is `Application/********-****-****-************/Documents`.

Function prototype

```
-(void)SetLogPath:(NSString*)logDir;
```

Parameter	Type	Description
logDir	NSString	Path

Sample code

```
[[ITMGContext GetInstance] SetLogPath:Path];
```

Callback Messages

Message list

Message	Description
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Indicates that PTT recording is completed.
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Indicates that PTT upload is completed.
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Indicates that PTT download is completed.
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Indicates that PTT playback is completed.
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Indicates that speech-to-text conversion is completed.

Data list

Message	Data	Sample
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	result; file_path	{"file_path":
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	result; file_path;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	result; file_path;file_id	{"file_id":"",'

ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	result; file_path	{"file_path":
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	result; text;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	result; file_path; text;file_id	{"file_id":"",'

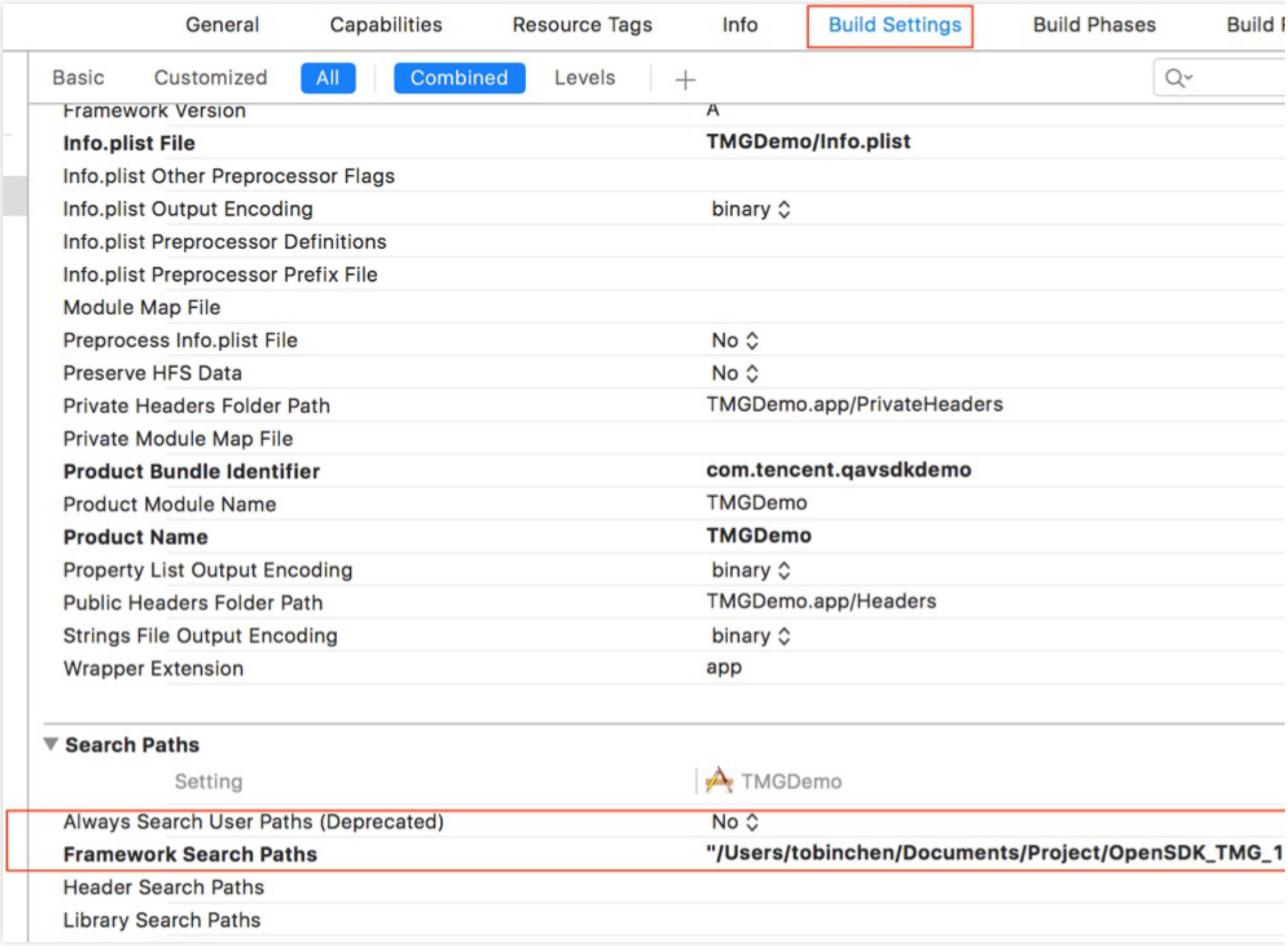
Project Export

최종 업데이트 날짜: : 2024-01-18 15:11:45

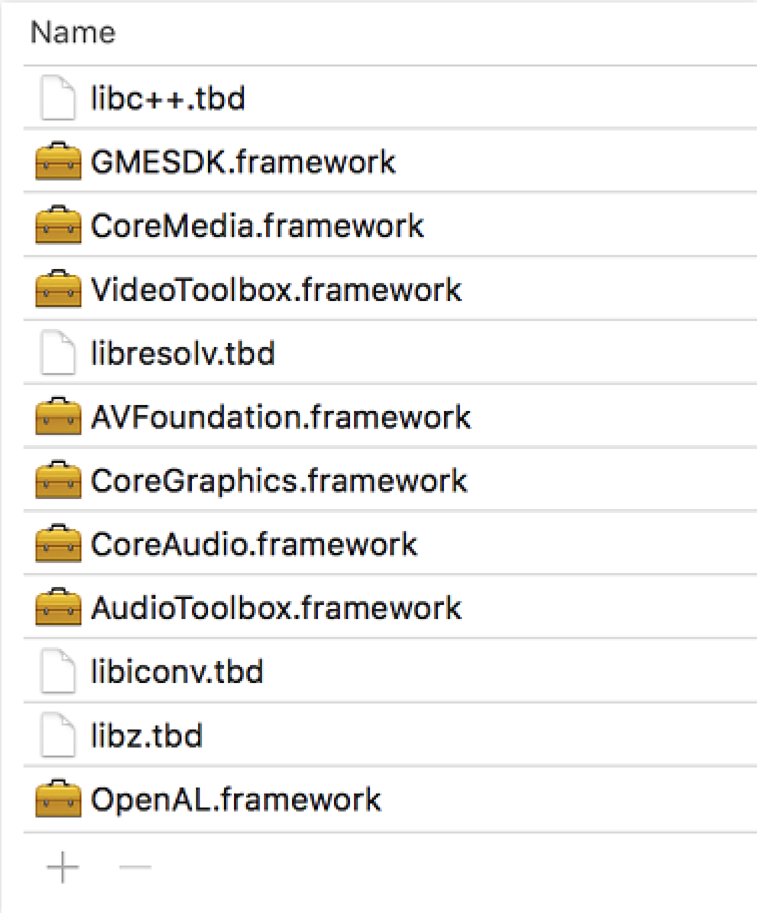
This document mainly describes the notes on exporting the iOS project so that the iOS developers can easily debug and integrate the APIs for Game Multimedia Engine (GME).

Export Configuration

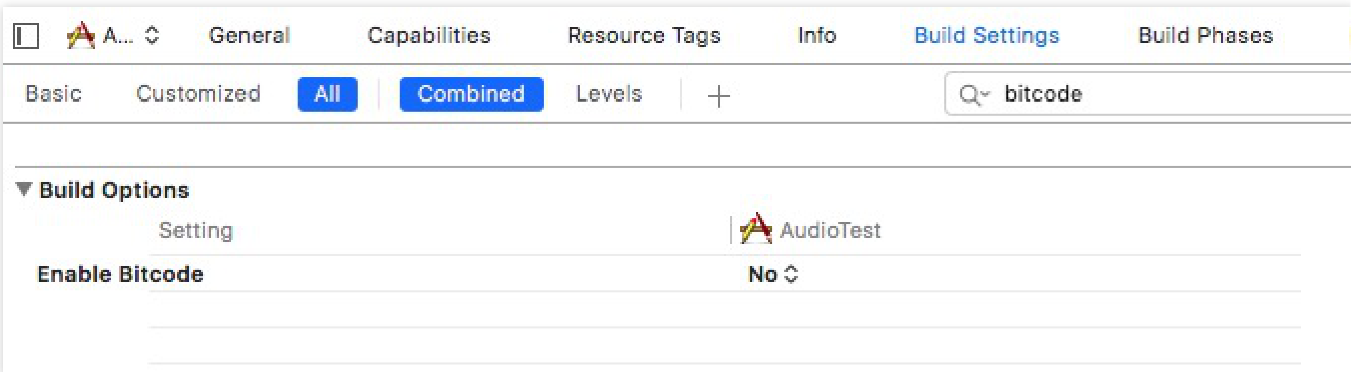
1. Add the following dependent library to **Xcode** > **Link Binary With Libraries** > **Build Setting** as needed, and set Framework Search Paths to point to the directory where the SDK resides, as shown below:



2. Add dependent libraries as shown below:



3. Bitcode should be supported by all class libraries that the project depends on. Bitcode is not supported by the SDK, so it can be disabled. To disable Bitcode, you only need to search for Bitcode under **Targets** > **Build Settings** and set the corresponding option to `NO`.



4. The GME SDK for iOS requires the following permissions:
Required background modes: Allows running in the background (optional).
Microphone Usage Description: Allows access to microphone.

GME 2.9 or Later

If the accessed SDK is on v2.9 or later, you need to configure it as instructed in iOS Project Upgrade Guide.

iOS Project Upgrade Guide

최종 업데이트 날짜: : 2024-01-18 15:11:45

Overview

The GME SDK has been upgraded to v2.9. To implement this upgrade, perform the following steps in your Xcode project:

Upgrade Directions

1. Download the SDK

In the new version, the dynamic libraries of the SDK are split into the following files:

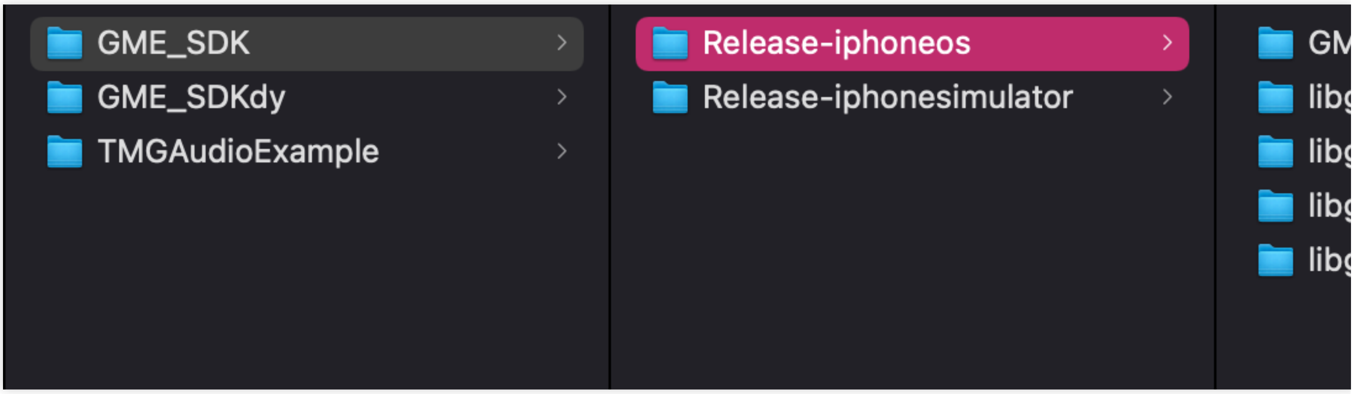
libgmefdkaac.framework

libgmeogg.framework

libgmelamemp3.framework

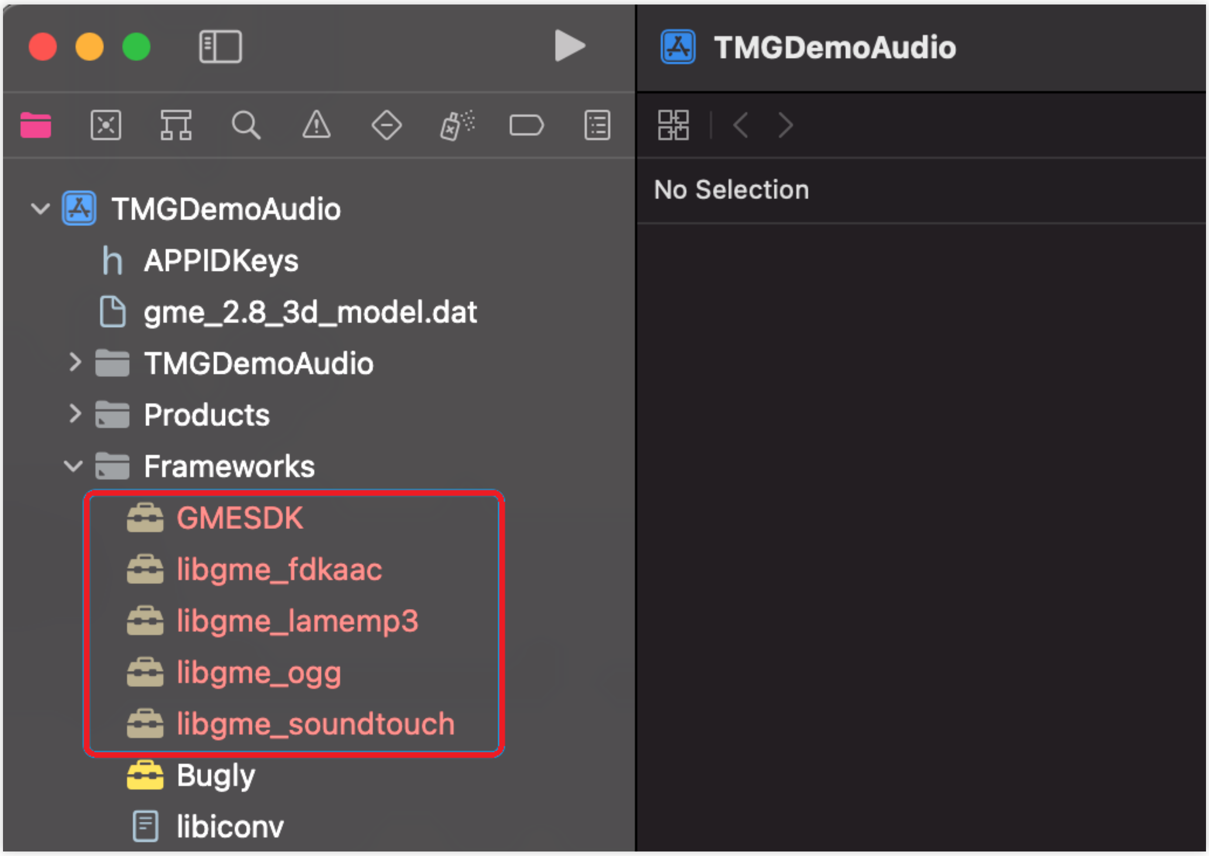
libgmesoundtouch.framework

Make sure that the downloaded SDK contains these files. After downloading, put them together with `GMESDK.framework` in the project directory. `Release-iphoneos` is the SDK file used for real devices, while `Release-iphonesimulator` is the SDK file used for simulators.



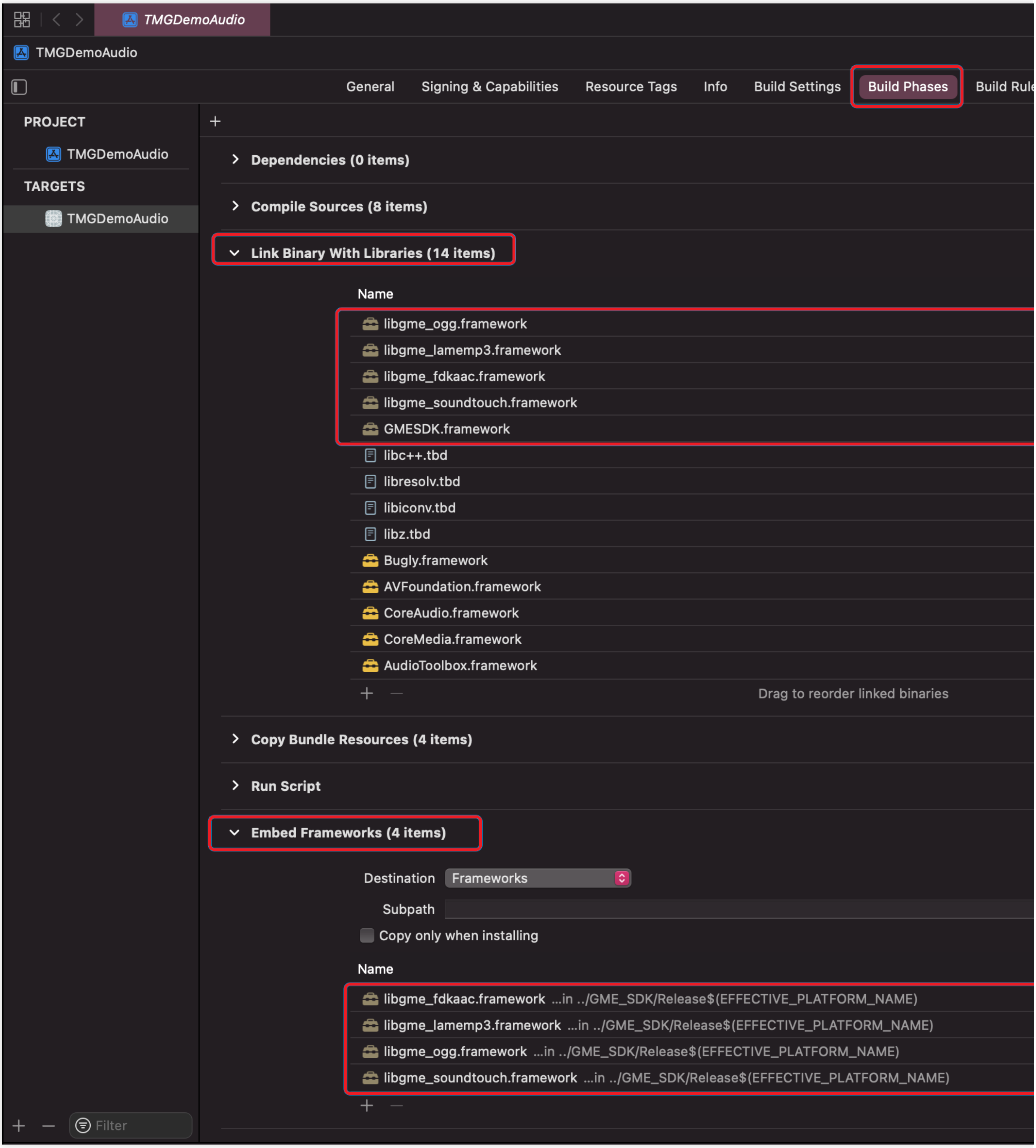
2. Import the SDK into the project

Import all frameworks into the project as shown below:



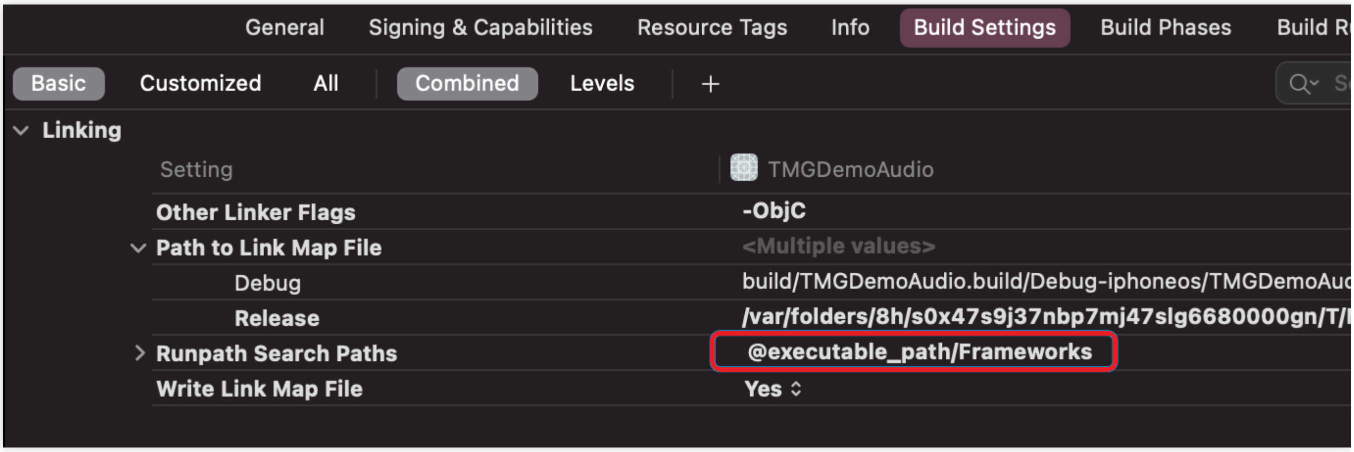
3. Configure frameworks and sign

1. In the Xcode project, click **Build Phases**, expand **Link Binary With Libraries**, and import all GME frameworks.
2. Expand **Embed Framework**, import all GME frameworks, and select **Code Sign On Copy**.



4. Modify the rpath

You need to add `@executable_path/Frameworks` in the `rpath`. If it has already been added, there is no need for modification.


General
Signing & Capabilities
Resource Tags
Info
Build Settings
Build Phases
Basic
Customized
All
Combined
Levels
Linking
Setting
TMGDemoAudio
Other Linker Flags
-ObjC
Path to Link Map File
<Multiple values>
Debug
build/TMGDemoAudio.build/Debug-iphoneos/TMGDemoAu
Release
/var/folders/8h/s0x47s9j37nbp7mj47slg6680000gn/T/
Runpath Search Paths
@executable_path/Frameworks
Write Link Map File
Yes

SDK for Android

Integrating SDK

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document describes how to integrate GME SDK into an Android project so that the Android developers can easily debug and integrate the APIs for Game Multimedia Engine (GME).

Preparing SDK

1. Download the applicable demo and SDK. For more information, see SDK Download Guide.
2. Decompress the obtained SDK resources.
3. The SDK development resources are in the `libs` folder.

Note:

You can run the SDK on Android 5.0 or later.

Configuration Guide

Method 1

1. Copy the `gmesdk.jar` file in the `libs` directory to the `libs` directory of the Android project.
2. **Copy the library files of the corresponding architecture based on the project requirements**. For example, if the project requires the armeabi-v7a architecture, you need to copy the library files in the `armeabi-v7a` directory to the `armeabi-v7a` directory in the project. If there is no `armeabi-v7a` directory in the project, create one.

Method 2

Place .so and .jar files in any folder in the project and specify the folder in `surceSets` .

Configuring the project

Add the code that imports the library to build.gradle under the App directory of the project.

```
sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
}
```

Voice Chat

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document describes how to integrate with and debug GME client APIs for the voice chat feature for Android.

Key Considerations for Using GME

GME provides the real-time voice chat service, voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

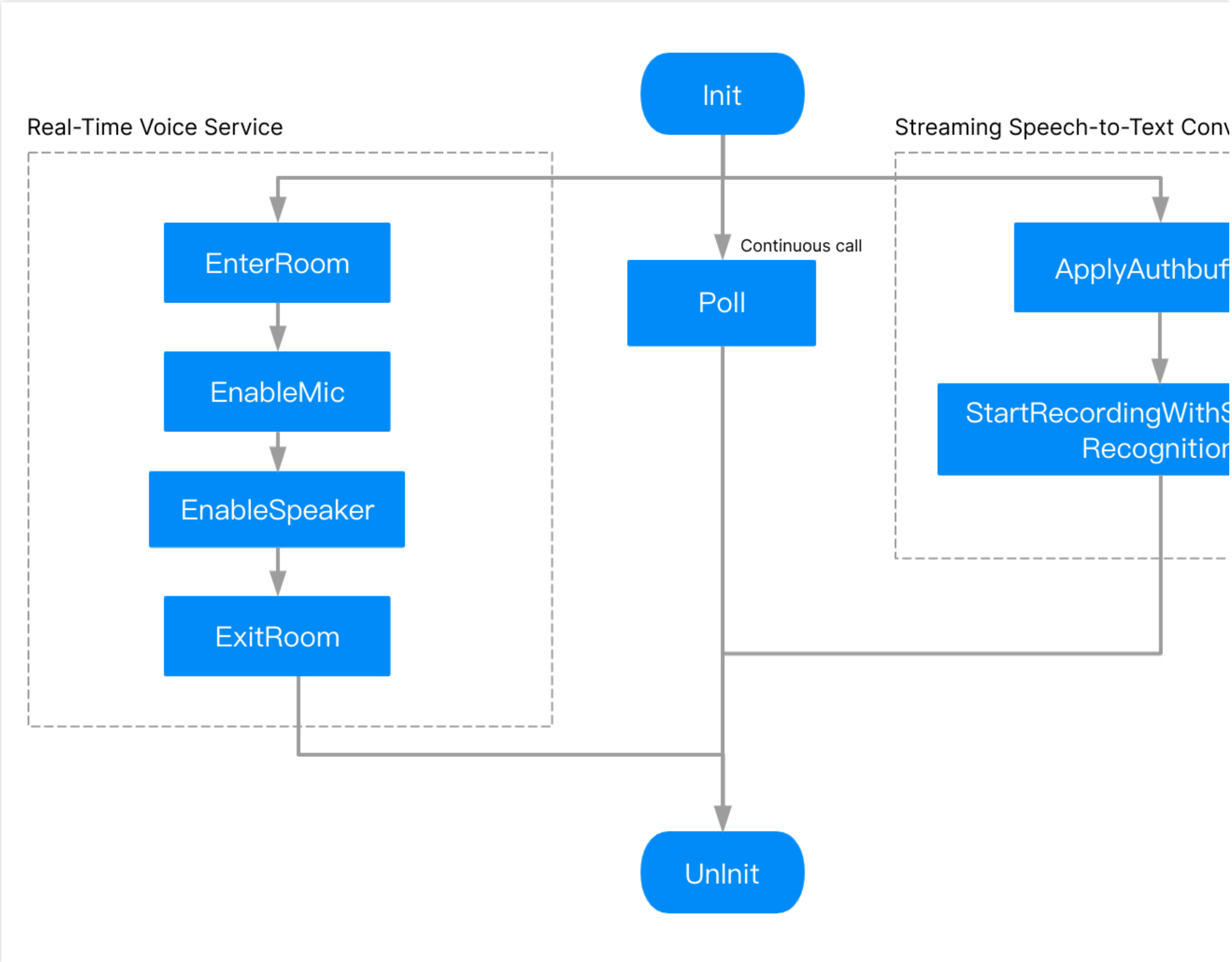
Notes

You have created a GME application and obtained the `AppID` and `Key` of the SDK as instructed in Activating Services.
You have **activated the voice chat, voice messaging, and speech-to-text services of GME**. For more information, see Activating Services.
Configure your project before using GME; otherwise, the SDK will not take effect.
After a GME API is called successfully, `QAVError.OK` will be returned with the value being `0`.
GME APIs should be called in the same thread.
The `Poll` API should be called periodically for GME to trigger event callbacks.
For detailed error codes, see Error Codes.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME
2. Calling Poll periodically to trigger callbacks
3. Entering a voice chat room
4. Turning on the mic
5. Turning on the speaker
6. Exiting the voice chat room
7. Uninitializing GME

Voice chat for Android class

Class	Description
ITMGContext	Core APIs
ITMGRoom	Room APIs
ITMGRoomManager	Room management APIs

ITMGAudioCtrl	Audio APIs
ITMGAudioEffectCtrl	Sound effect and accompaniment APIs

Core APIs

API	Description
Init	Initializes GME.
Poll	Triggers an event callback.
Pause	Pauses the system.
Resume	Resumes the system.
Uninit	Uninitializes GME.

Note:
If you need to switch the account, please call `UnInit` to uninitialize the SDK. No fee is incurred for calling Init API.

Getting singleton

To use the voice feature, get the `ITMGContext` object first.

Sample code

```
import com.tencent.TMG.ITMGContext;
ITMGContext.getInstance(this);
```

Registering callback

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages.

API prototype

```
static public abstract class ITMGDelegate {
    public void OnEvent(ITMG_MAIN_EVENT_TYPE type, Intent data){}
}
```

Override this callback function in the constructor to process the parameters of the callback.

Parameter	Type	Description

type	ITMGContext.ITMG_MAIN_EVENT_TYPE	Event type in the callback response
data	Intent message type	Callback message, i.e., event data

Sample code

```
private ITMGContext.ITMGDelegate itmgDelegate = null;
itmgDelegate = new ITMGContext.ITMGDelegate() {
    @Override
    public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
        if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM == typ
        {
            // Analyze the returned data
            int nErrCode = data.getIntExtra("result" , -1);
            String strErrMsg = data.getStringExtra("error_info");
                }
        }
}
```

Register the callback function to the SDK before room entry.

API prototype

```
public abstract int SetTMGDelegate(ITMGDelegate delegate);
```

Parameter	Type	Description
delegate	ITMGDelegate	SDK callback function

Sample code

```
ITMGContext.GetInstance(this).SetTMGDelegate(itmgDelegate);
```

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice chat, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
public abstract int Init(String sdkAppId, String openId);
```

Parameter	Type	Description
sdkAppId	String	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openId	String	`openID` can only be in Int64 type, which is passed in after being converted to a `const char*` . You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
QAVError.OK= 0	SDK was initialized successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Check whether the SDK file is complete. We recommend that you delete it and then import the SDK again.

Notes on 7015 error code

The error code 7015 is identified based on the MD5 value. If this error is reported during integration, check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party enhancement, the Unity packaging mechanism, and other factors, the library file MD5 will be affected, resulting in misjudgment. Therefore, **ignore this error in the logic for official releases**, and avoid displaying it on the UI.

Sample code

```
String sdkAppID = "14000xxxxx";
String openID = "100";
int ret = 0;
// After the user agrees to the application's privacy policy, initialize the SDK at
//ret = 0: The user agrees to the application's privacy policy
//ret = 1: The user does not agree to the application's privacy policy
// If the user does not agree to the privacy policy, change `ret` to a value other
if(ret != 0){
    Log.e(TAG,"The user does not agree to the application's privacy policy");
}else{
        ITMGContext.GetInstance(this).Init(sdkAppId, openId);
}
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. `Poll` is the message pump of GME, and the `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.

Refer to the `EnginePollHelper.java` file in SDK Download Guide.

Call the `Poll` API periodically

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
public abstract int Poll();
```

Sample code

```
private Handler mhandler = new Handler();private Runnable mRunnable = new Runnable(
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. If you do not need the audio played back in the background in the room, call `Pause` API to pause the GME service.

API prototype

```
public abstract int Pause();
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
public abstract int Resume();
```

Uninitializing SDK

This API is used to uninitialize the SDK. **If the game account is bound to** `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.

Note:

If the end user revokes the permission granted to the SDK to process the personal information, you can call the `Uninit` API to stop using the SDK features and stop collecting and close the user data used by the features.

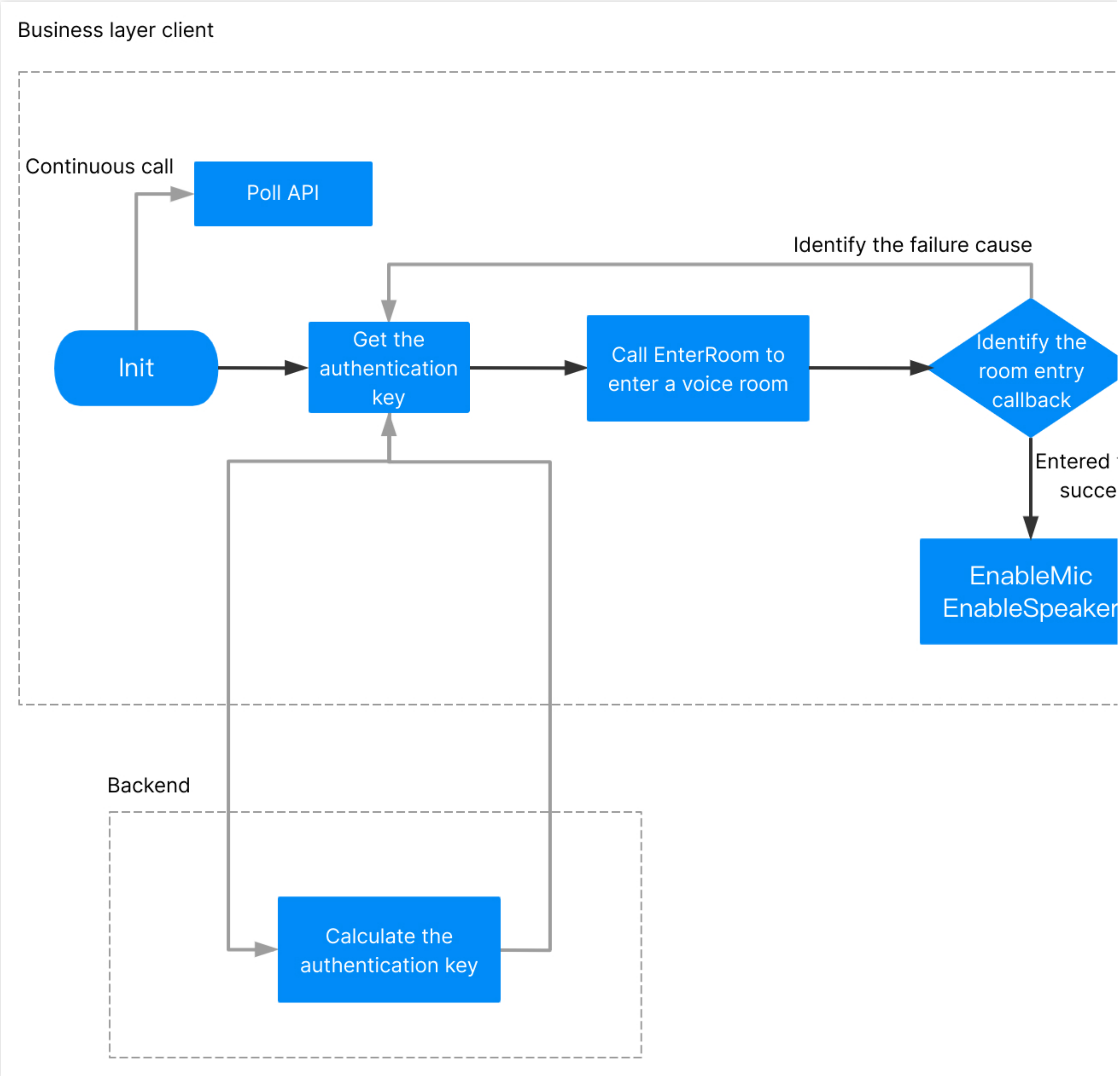
API prototype

```
public abstract int Uninit();
```

Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, see Sound and Audio.



API	Description
GenAuthBuffer	Calculates the local authentication key.
EnterRoom	Enters a room.
ExitRoom	Exits a room.
IsRoomEntered	Determines whether room entry is successful.
SwitchRoom	Switches the room quickly.
StartRoomSharing	Cross-room Co-anchoring

Local authentication key calculation

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.

API prototype

```
AuthBuffer public native byte[] genAuthBuffer(int sdkAppId, String roomId, String o
```

Parameter	Type	Description
appId	int	`AppId` from the Tencent Cloud console
roomId	String	Room ID, which can contain up to 127 characters.
openId	String	User ID, which is the same as `OpenId` during initialization.
key	String	Permission key from the Tencent Cloud console.

Sample code

```
import com.tencent.av.sig.AuthBuffer;// Header file
byte[] authBuffer = AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(sdkAppI
```

Entering a room

This API is used to enter a room with the generated authentication information. The mic and speaker are not enabled by default after room entry.

Note:

If the room entry callback result is `0`, the room entry is successful. If `0` is returned from the `EnterRoom` API, it doesn't necessarily mean that the room entry is successful.

The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will become smooth sound quality. Only when a member in the room calls the `ChangeRoomType` API, the audio type of the room will be changed.

API prototype

```
public abstract int EnterRoom(String roomID, int roomType, byte[] authBuffer);
```

Parameter	Type	Description

roomId	String	Room ID, which can contain up to 127 characters.
roomType	int	Room type. We recommend that you enter `ITMG_ROOM_TYPE_FLUENCY` . For more information on room audio types, see Sound Quality.
authBuffer	byte[]	Authentication key

Sample code

```
ITMGContext.GetInstance(this).EnterRoom(roomId,roomType, authBuffer);
```

Callback for room entry

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function for callback and processing. A successful callback means that the room entry is successful, and the billing starts.

Billing references:

Purchase Guide

Billing

Will the billing continue if the client is disconnected from the server when using the voice chat?

Function prototype

```
private ITMGContext.ITMGDelegate itmgDelegate = null;
itmgDelegate= new ITMGContext.ITMGDelegate() {
            @Override
             public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data
                }
};
```

Sample code

Sample code for processing the callback, including room entry and network disconnection events.

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM == type)
        {
              // Analyze the returned data
            int nErrCode = data.getIntExtra("result" , -1);
            String strErrMsg = data.getStringExtra("error_info");

            if (nErrCode == AVError.AV_OK)
            {
```



```
                //Entered room successfully, and you can proceed with your operatio
                ScrollView_ShowLog("EnterRoom success");
                Log.i(TAG,"EnterRoom success!");
            }
            else
            {
                //If you fail to enter the room, you need to analyze the returned e
                ScrollView_ShowLog("EnterRoom fail :" + strErrMsg);
                Log.i(TAG,"EnterRoom fail!");
            }
        }
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT == ty
        {
            //waiting timeout, please check your network
        }
    }
```

Data details

Message	Data	Example
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnected callback prompt `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` . At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` . When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` .

Error codes

Error Code	Cause and Suggested Solution
7006	Authentication failed. Causes: `AppID` doesn't exist or is incorrect. An error occurred while authenticating `authbuff` . Authentication expired. `OpenId` is invalid.
7007	The user was already in another room.
1001	The user was already in the process of entering a room but repeated this operation. We

	recommend that you not call the room entry API until the room entry callback is returned.
1003	The user was already in the room and called the room entry API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally.

Exiting a room

This API is used to exit the current room. It is an async API. The returned value `AV_OK` indicates a successful async delivery. If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API; instead, you can directly call the `EnterRoom` API.

API prototype

```
public abstract int ExitRoom();
```

Sample code

```
ITMGContext.GetInstance(this).ExitRoom();
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM`.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_EXIT_ROOM == type)
        {
            // Receive the event of successful room exit
        }
}
```

Data details

Message	Data	Example
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}

Determining whether user has entered room

This API is used to determine whether the user has entered a room. A value in bool type will be returned. Do not call this API during room entry.

API prototype

```
public abstract boolean IsRoomEntered();
```

Sample code

```
ITMGContext.GetInstance(this).IsRoomEntered();
```

Switching room

User can call this API to quickly switch the voice chat room after entering the room. After the room is switched, the device is not reset, that is, if the microphone is already enabled in this room, the microphone will keep enabled after the room is switched.

The callback for quickly switching rooms is `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM` , and the fields are `error_info` and `result` .

API prototype

```
public abstract int SwitchRoom(String targetRoomID, byte[] authBuffer);
```

Type descriptions

Parameter	Type	Description
targetRoomID	String	ID of the room to enter
authBuffer	byte[]	Generates a new authentication key with the ID of the room to enter

Callback sample code

```
if(ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM == type) {
int result = data.getIntExtra("result", 1);
String errorInfo = data.getStringExtra("error_info");
if (result == 0) {
            Toast.makeText(getActivity(), "switch room success.", Toast.LENGTH_SHOR
            }
else {
            Toast.makeText(getActivity(), "switch room failed.. error info=" + erro
            }
```



```
}
```

Cross-room mic connection

Call this API to connect the microphones across rooms after the room entry. After the call, the local user can communicate with the target OpenID user in the target room. The target room should be of the same type as the local room.

Example

User a is in room A, user b is in room B, and user a can talk with b through the cross-room API. When user c in room A speaks, users b and d in room B cannot hear. User c in room A can hear only the voice in room A and the voice of user b in room B but not other users in room B.

API prototype

```
/// <summary> Enable the room sharing, and connect the mic of the OpenID in another
public abstract int StartRoomSharing(String targetRoomID, String targetOpenID, byte
/// <summary> Stop the enabled room sharing.</summary>
public abstract int StopRoomSharing();
```

Type descriptions

Parameter	Type	Description
targetRoomID	String	ID of the room to connect mic
targetOpenID	String	Target `OpenID` to connect mic
authBuffer	byte[]	Reserved flag. You just need to enter NULL.

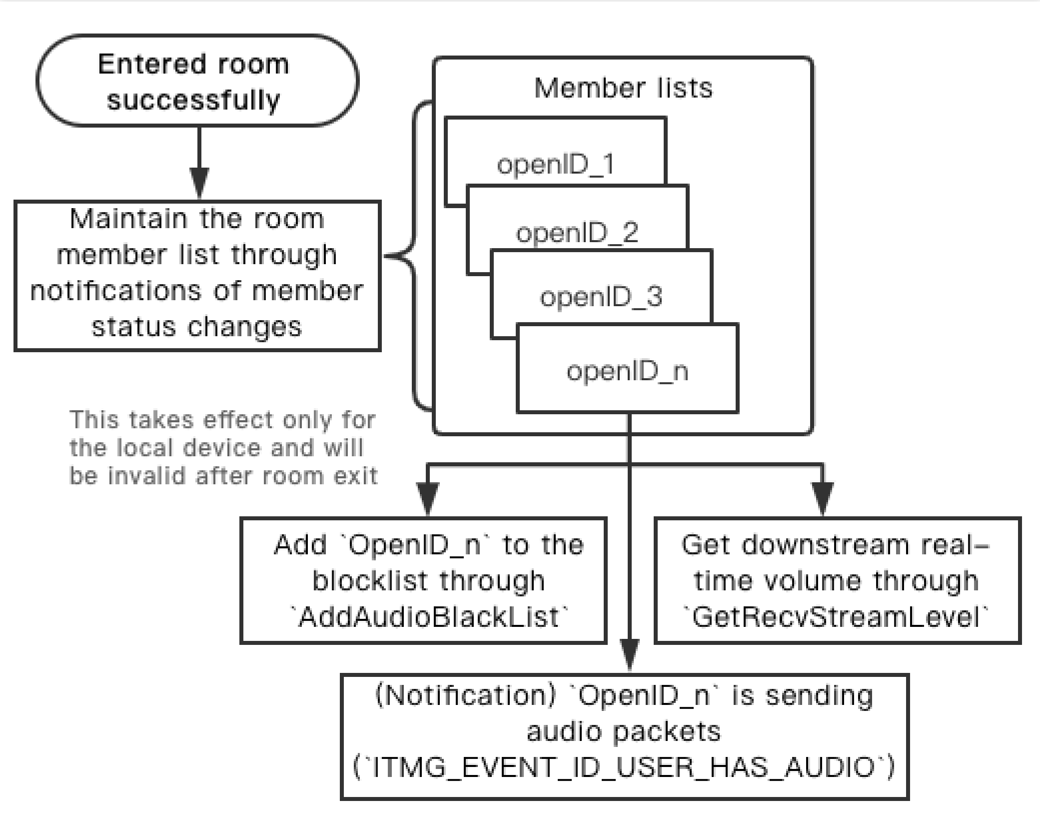
Sample code

```
if (mSwtichRoomShareStart.isChecked())
    {
        String strRoomID = mEditRoomShareRoomID.getText().toString();
        String strOpenID = mEditRoomShareOpenID.getText().toString();
        int nRet = ITMGContext.GetInstance(getActivity()).GetRoom().StartRoomSharin
        if (nRet != 0)
            {
                Toast.makeText(getActivity(), String.format("StartRoomSharing faile
            }else
            {
                int nRet = ITMGContext.GetInstance(getActivity()).GetRoom().StopRoo
                    if (nRet != 0)
```

```
                                {
                                    Toast.makeText(getActivity(), String.format("StopRoomSh
                                }
                        }
}
```

Room Status Maintenance

APIs in this section are used to display speaking members and members entering or exiting the room and mute a member in the room at the business layer.



API/Notification	Description
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The member status changed.
AddAudioBlackList	Mutes a member in the room.
RemoveAudioBlackList	Unmutes a member.

Notifications of member room entry and speaking status

This API is used to obtain the user speaking in the room and display it in the UI, and to send a notification when someone entering or exiting the room.

Notification for this event will be sent only when the status changes. To get member status in real time, cache the notification when it is received at a higher layer. The event message is `ITMG_MAIN_EVNET_TYPE_USER_UPDATE` , where the data contains `event_id` and `user_list` . The event message will be identified in the `OnEvent` function.

Notifications for audio events are subject to a threshold, and a notification will be sent only when this threshold is exceeded. The notification "A member has stopped sending audio packets" will be sent if no audio packets are received in more than two seconds.

event_id	Description	Maintenance
ITMG_EVENT_ID_USER_ENTER	Return the `openid` of the member entering the room.	Member list
ITMG_EVENT_ID_USER_EXIT	Return the `openid` of the member exiting the room.	Member list
ITMG_EVENT_ID_USER_HAS_AUDIO	Return the `openid` of the member sending audio packets in the room. This event can be used to determine whether a user is speaking and display the voiceprint effect.	Chat member list
ITMG_EVENT_ID_USER_NO_AUDIO	Return the `openid` of the member stopping sending audio packets in the room.	Chat member list

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_USER_UPDATE == type)
        {
        // Update member status
        int nEventID = data.getIntExtra("event_id", 0);
        String[] openIdList =data.getStringArrayExtra("user_list");
         switch (nEventID)
             {
                  case ITMG_EVENT_ID_USER_ENTER:
                // A member enters the room
                break;
                  case ITMG_EVENT_ID_USER_EXIT:
                // A member exits the room
                      break;
                  case ITMG_EVENT_ID_USER_HAS_AUDIO:
                      // A member sends audio packets
                      break;
                  case ITMG_EVENT_ID_USER_NO_AUDIO:
                      // A member stops sending audio packets
```

```
                        break;
                    default:
                        break;
                }
            }
}
```

Data details

Message	Data	Example
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	event_id; user_list	{"event_id":0,"user_list":""}

Muting a member in the room

This API is used to add an ID to the audio data blocklist. This operation blocks audio from someone and only applies to the local device without affecting other devices. The returned value `0` indicates that the call is successful.

Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B.

If B blocks neither A nor C, B can hear both of them.

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

API prototype

```
public abstract int AddAudioBlackList(String openId);
```

Parameter	Type	Description
openId	String	`openid` of the user to be blocked

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().AddAudioBlackList(openId);
```

Unmuting

This API is used to remove an ID from the audio data blacklist. A returned value of 0 indicates the call is successful.

API prototype

```
public abstract int RemoveAudioBlackList(String openId);
```

Parameter	Type	Description
openId	String	User `openid` to be unblocked

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().RemoveAudioBlackList(openId);
```

Voice Chat Capturing APIs

The voice chat APIs can only be called after SDK initialization and room entry.
When the user clicks the button of enabling/disabling the mic or speaker on the UI, we recommend that you call the `EnableMic` or `EnableSpeaker` API.
When the user enters a voice chat room, enabling/disabling a capturing device will restart both capturing and playback devices. If the application is playing back background music, it will also be interrupted. Playback won't be interrupted if the mic is enabled/disabled through control of upstreaming/downstreaming. **Calling method: Call** `EnableAudioCaptureDevice(true)` and `EnableAudioPlayDevice(true)` once after room entry, and call `EnableAudioSend/Recv` to send/receive audio streams when Enable/Disable Mic is clicked.

API	Description
EnableMic	Enables/Disables the mic.
GetMicState	Gets the mic status.
EnableAudioCaptureDevice	Enables/Disables the capturing device.
IsAudioCaptureDeviceEnabled	Gets the capturing device status.
EnableAudioSend	Enables/Disables audio upstreaming.
IsAudioSendEnabled	Gets the audio upstreaming status.
GetMicLevel	Gets the real-time mic volume level.
GetSendStreamLevel	Gets the real-time audio upstreaming volume level.
SetMicVolume	Sets the mic volume level.
GetMicVolume	Gets the mic volume level.

Enabling or disabling mic

This API is used to enable/disable the mic. The mic and speaker are not enabled by default after room entry. `EnableMic` is equivalent to using `EnableAudioCaptureDevice` and `EnableAudioSend` together. **If accompaniment is used, call this API as instructed in** Accompaniment in Voice Chat.

API prototype

```
public abstract int EnableMic(boolean isEnabled);
```

Parameter	Type	Description
isEnabled	boolean	To enable the mic, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
// Turn on mic
ITMGContext.GetInstance(this).GetAudioCtrl().EnableMic(true);
```

Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

API prototype

```
public abstract int GetMicState();
```

Sample code

```
int micState = ITMGContext.GetInstance(this).GetAudioCtrl().GetMicState();
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.
This API can only be called after room entry. The device will be disabled automatically after room exit.
Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

API prototype

```
public abstract int EnableAudioCaptureDevice(boolean isEnabled);
```

Parameter	Type	Description

isEnabled	boolean	To enable the capturing device, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable capturing device
ITMGContext.GetInstance(this).GetAudioCtrl().EnableAudioCaptureDevice(true);
```

Getting the capturing device status

This API is used to get the status of a capturing device.

API prototype

```
public abstract boolean IsAudioCaptureDeviceEnabled();
```

Sample code

```
bool IsAudioCaptureDevice = ITMGContext.GetInstance(this).GetAudioCtrl().IsAudioCap
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain mute. For more information on how to enable/disable the capturing device, see the `EnableAudioCaptureDevice` API.

API prototype

```
public abstract int EnableAudioSend(boolean isEnabled);
```

Parameter	Type	Description
isEnabled	boolean	To enable audio upstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().EnableAudioSend(true);
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

API prototype

```
public abstract boolean IsAudioSendEnabled();
```

Sample code

```
bool IsAudioSend = ITMGContext.GetInstance(this).GetAudioCtrl().IsAudioSendEnabled(
```

Getting the real-time mic volume

This API is used to get the real-time mic volume. An int-type value in the range of 0-100 will be returned. It is recommended to call this API once every 20 ms.

API prototype

```
public abstract int GetMicLevel();
```

Sample code

```
int micLevel = ITMGContext.GetInstance(this).GetAudioCtrl().GetMicLevel();
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume. An int-type value in the range of 0-100 will be returned.

API prototype

```
ITMGContext TMGAudioCtrl int GetSendStreamLevel()
```

Sample code

```
int Level = ITMGContext.GetInstance(this).GetAudioCtrl().GetSendStreamLevel();
```

Setting the mic software volume

This API is used to set the mic volume level. The corresponding parameter is `volume` , which is equivalent to attenuating or gaining the captured sound.

API prototype

```
public abstract int SetMicVolume(int volume);
```

Parameter	Type	Description
volume	int	Value range: 0-200. Default value: `100` . `0` indicates that the audio is muted, while `100` indicates that the volume level remains unchanged.

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().SetMicVolume(volume);
```

Getting the mic software volume

This API is used to obtain the microphone volume. An "int" value is returned. Value 101 represents API SetMicVolume has not been called.

API prototype

```
public abstract int GetMicVolume();
```

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().GetMicVolume();
```

Voice Chat Playback APIs

API	Description
EnableSpeaker	Enables/Disables the speaker.
GetSpeakerState	Gets the speaker status.
EnableAudioPlayDevice	Enables/Disables the playback device.
IsAudioPlayDeviceEnabled	Gets the playback device status.
EnableAudioRecv	Enables/Disables audio downstreaming.
IsAudioRecvEnabled	Gets the audio downstreaming status.
GetSpeakerLevel	Gets the real-time speaker volume level.
GetRecvStreamLevel	Gets the real-time downstreaming audio levels of other members in the room.

SetSpeakerVolume	Sets the speaker volume level.
GetSpeakerVolume	Gets the speaker volume level.

Enabling or disabling speaker

This API is used to enable/disable the speaker. `EnableSpeaker` is equivalent to using `EnableAudioPlayDevice` and `EnableAudioRecv` together. **If accompaniment is used, call this API as instructed in** Accompaniment in Voice Chat.

API prototype

```
public abstract int EnableSpeaker(boolean isEnabled);
```

Parameter	Type	Description
isEnabled	boolean	To disable the speaker, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
// Turn on the speaker
ITMGContext.GetInstance(this).GetAudioCtrl().EnableSpeaker(true);
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, and 1 is on.

API prototype

```
public abstract int GetSpeakerState();
```

Sample code

```
int micState = ITMGContext.GetInstance(this).GetAudioCtrl().GetSpeakerState();
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

API prototype

```
public abstract int EnableAudioPlayDevice(boolean isEnabled);
```

Parameter	Type	Description
isEnabled	boolean	To disable the playback device, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
// Enable the playback device
ITMGContext.GetInstance(this).GetAudioCtrl().EnableAudioPlayDevice(true);
```

Getting the playback device status

This API is used to get the status of a playback device.

API prototype

```
public abstract boolean IsAudioPlayDeviceEnabled();
```

Sample code

```
bool IsAudioPlayDevice = ITMGContext.GetInstance(this).GetAudioCtrl().IsAudioPlayDe
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain muted. For more information on how to enable/disable the playback device, see the `EnableAudioPlayDevice` API.

API prototype

```
public abstract int EnableAudioRecv(boolean isEnabled);
```

Parameter	Type	Description
isEnabled	boolean	To enable audio downstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().EnableAudioRecv(true);
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

API prototype

```
public abstract boolean IsAudioRecvEnabled();
```

Sample code

```
bool IsAudioRecv = ITMGContext.GetInstance(this).GetAudioCtrl().IsAudioRecvEnabled(
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume. An int-type value will be returned to indicate the volume. It is recommended to call this API once every 20 ms.

API prototype

```
public abstract int GetSpeakerLevel();
```

Sample code

```
int SpeakLevel = ITMGContext.GetInstance(this).GetAudioCtrl().GetSpeakerLevel();
```

Getting the real-time downstreaming audio levels of other members in room

This API is used to get the real-time audio downstreaming volume of other members in the room. An int-type value will be returned. Value range: 0-200.

API prototype

```
public abstract int GetRecvStreamLevel(String openId);
```

Parameter	Type	Description
openId	String	`openId` of another member in the room

Sample code

```
int Level = ITMGContext.GetInstance(this).GetAudioCtrl().GetRecvStreamLevel(openId)
```

Dynamically setting the volume of a member of the room

This API is used to set the volume of a member in the room. It takes effect only on the local.

API prototype

```
public abstract int SetSpeakerVolumeByOpenID(String openId, int volume);
```

Parameter	Type	Description
openId	String	`OpenID` of the target user
volume	int	Percentage. Recommended value range: 0-200. Default value: `100` .

Sample code

Executed statements

```
// Lower the volume of 123333 to 80%
String strOpenID = "1233333";
int nOpenVolume = Integer.valueOf(80);
int nRet = ITMGContext.GetInstance(getActivity()).GetAudioCtrl().SetSpeakerVolumeBy
if (nRet != 0)
{
  // Toast error occured
}
else
{
  // Toast set successfully
}
```

Getting volume percentage

Call this API to get the volume set by SetSpeakerVolumeByOpenID

API prototype

```
public abstract int GetSpeakerVolumeByOpenID(String openId);
```

Parameter	Type	Description
openId	String	`OpenID` of the target user

Returned values

API returns volume percentage set by OpenID, where 100 is by default.

Setting the speaker volume

This API is used to set the speaker volume.

API prototype

```
public abstract int SetSpeakerVolume(int volume);
```

Parameter	Type	Description
volume	int	Volume level. Value range: 0-200. Default value: `100`. `0` indicates that the audio is muted, while `100` indicates that the volume level remains unchanged.

Sample code

```
int speVol = (int)(value * 100);ITMGContext.GetInstance(this).GetAudioCtrl().SetSpe
```

Getting the speaker volume

This API is used to get the speaker volume. An int-type value will be returned to indicate the volume. 101 indicates that the `SetSpeakerVolume` API has not been called.
"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

API prototype

```
public abstract int GetSpeakerVolume();
```

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().GetSpeakerVolume();
```

Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

API prototype

```
public abstract int EnableLoopBack(boolean enable);
```

Parameter	Type	Description
enable	boolean	Specifies whether to enable in-ear monitoring.

Sample code

```
ITMGContext.GetInstance(this).GetAudioCtrl().EnableLoopBack(true);
```

Getting user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, see the `EnterRoom` API.

API prototype

```
public abstract int GetRoomType();
```

Sample code

```
ITMGContext.GetInstance(this).GetRoom().GetRoomType();
```

Getting the room ID

This API is used to get the voice chat room ID and can be called only after a successful room entry. A string will be returned.

API prototype

```
public abstract String GetRoomID();
```

Modifying user's room audio type

This API is used to modify a user's room audio type. For the result, please see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

API prototype

```
public abstract int ChangeRoomType(int nRoomType);
```

Parameter	Type	Description
nRoomType	int	Room type to be switched to. For room audio types, see the `EnterRoom` API.

Sample code

```
ITMGContext.GetInstance(this).GetRoom().ChangeRoomType(nRoomType);
```

Callback for modifying the room type

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result` , `error_info` , and `new_room_type` . The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	The existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	A user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	A user is already in the room, and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	A room member calls the `ChangeRoomType` API to request a change of the room audio type.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {        if
```

Data details

Message	Data	Exa
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result;error_info;new_room_type;subEventType	{"e

The monitoring event of room call quality

This is the quality monitoring event, which is triggered once every two seconds after room entry, and its message is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY` . The returned parameters include `weight` , `loss` , and `delay` , which are as detailed below:

Parameter	Type	Description
weight	int	Value range: 1–50. `50` indicates excellent sound quality, `1` indicates very poor (barely usable) sound quality, and `0` represents an initial meaningless value. Generally, if the value is below 30, you can remind users that the network is poor and recommend them to switch the network.
Loss	double	Upstream packet loss rate
Delay	int	Voice chat delay in ms

Getting version number

This API is used to get the SDK version number for analysis.

API prototype

```
public abstract String GetSDKVersion();
```

Sample code

```
ITMGContext.GetInstance(this).GetSDKVersion();
```

Checking mic permission

This API is used to return the mic permission status.

Function prototype

```
public abstract ITMG_RECORD_PERMISSION  CheckMicPermission();
```

Parameter description

Parameter	Value	Description
ITMG_PERMISSION_GRANTED	0	The mic permission is granted.
ITMG_PERMISSION_Denied	1	Microphone disabled.

ITMG_PERMISSION_NotDetermined	2	No authorization box has been popped up to request the permission.
ITMG_PERMISSION_ERROR	3	An error occurred while calling the API.

Sample code

```
ITMGContext.GetInstance(this).CheckMicPermission();
```

Checking mic status

Function prototype

```
public abstract ITMG_CHECK_MIC_STATUS CheckMic();
```

Returned value handling

Returned Value	Description	Handling
ITMG_CHECK_MIC_STATUS_AVAILABLE = 0	Normally available	No handling required
ITMG_CHECK_MIC_STATUS_NO_GRANTED = 2	Access not obtained/denied	The access permission needs to be obtained before the mic is enabled.
ITMG_CHECK_MIC_STATUS_INVALID_MIC = 3	No device available	Generally, this error will be reported on PCs when no mics are available. Prompt the user to insert a headset or mic.
ITMG_CHECK_MIC_STATUS_NOT_INIT = 5	Not initialized	Call `EnableMic` after `Init` .

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
public abstract int SetLogLevel(int levelWrite, int levelPrint);
```

Parameter description

Parameter	Type	Description

levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO` .
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: `TMG_LOG_LEVEL_ERROR` .

`ITMG_LOG_LEVEL` description:

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContext.GetInstance(this).SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path, and needs to be called before initialization. The default path is /sdcard/Android/data/xxx.xxx.xxx/files.

API prototype

```
public abstract int SetLogPath(String logDir);
```

Parameter	Type	Description
logDir	String	Path

Sample code

```
ITMGContext.GetInstance(this).SetLogPath(path);
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

API prototype

```
public abstract String GetQualityTips();
```

Sample code

```
ITMGContext.GetInstance(this).GetRoom().GetQualityTips();
```

Callback Messages

Message	Description	Data
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room.	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room.	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected due to a network or another issue.	result; error
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated.	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	The reconnection to the room started.	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	The reconnection	result; error

	to the room succeeded.	
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched.	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status was changed.	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started.	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped.	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker was changed.	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A new speaker was added.	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker was lost.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A new mic was added.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic was lost.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic was changed.	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room quality changed.	weight; loss delay

ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Voice message recording was completed.	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Voice message upload was completed.	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Voice message download was completed.	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Voice message playback was completed.	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast speech-to-text conversion was completed.	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed.	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	Streaming speech-to-text conversion is in progress.	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed.	result; text;file_id

ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed.	result; text;file_id

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document describes how to integrate with and debug the GME APIs to implement speech-to-text service for Android.

Note:

This document applies to GME SDK version 2.9.

Key Considerations for Using GME

GME provides voice chat service, voice messaging and speech-to-text services and they rely on core APIs such as Init and Poll.

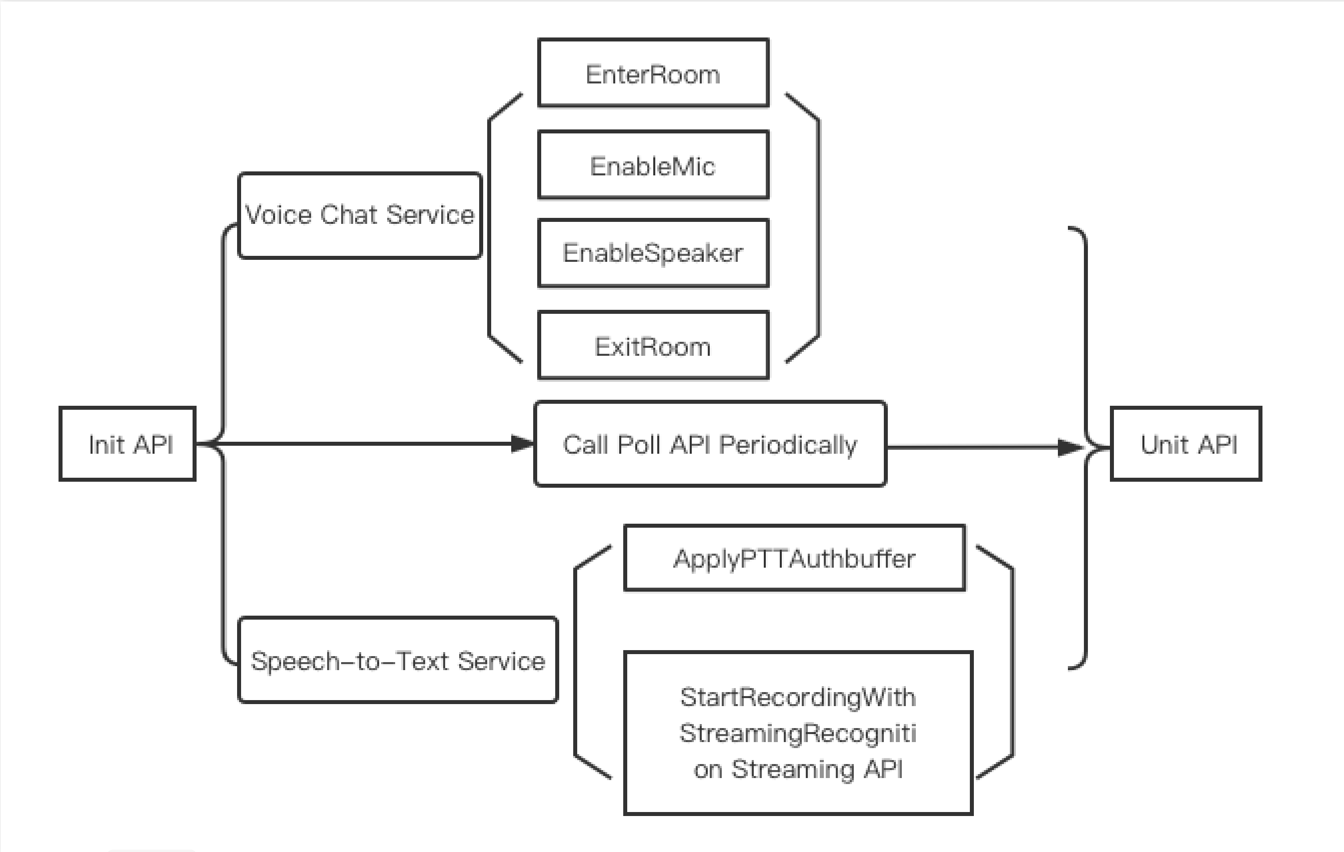
Note:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.

Non-streaming speech-to-text API **SpeechToText()**: There can be up to 10 concurrent requests per account.

Streaming speech-to-text API **StartRecordingWithStreamingRecognition()**: There can be up to 50 concurrent requests per account.

Voice chat streaming speech-to-text API **StartRealTimeASR()**: There can be up to 50 concurrent requests per account.



Notes on the `Init` API

If you need to use voice chat and voice messaging services at the same time, **you only need to call** `Init` API once.

The billing will not start after initialization. Receiving or sending a voice message in speech-to-text service is counted as a voice message DAU.

Directions

1. Initializing GME
2. Calling Poll periodically to trigger callbacks
3. Initializing authentication
4. Starting streaming speech-to-text conversion
5. Stopping recording
6. Uninitializing GME

Important notes

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being `0` .

GME APIs should be called in the same thread.

The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error codes, see Error Codes.

Voice message for Android class

Class	Description
ITMGContext	Core APIs
ITMGPTT	Voice messaging and speech-to-text APIs

Core APIs

Before the initialization, the SDK is in the uninitialized status, and **you need to initialize it through the** `Init` API before you can use the voice chat and speech-to-text services.

Call the `Init` API before calling any APIs of GME.

If you have any questions when using the service, see General.

API	Description
Init	Initializes GME.
Poll	Triggers an event callback.
Pause	Pauses the system.
Resume	Resumes the system.
Uninit	Uninitializes GME.

Note:

If you need to switch the account, please call `UnInit` to uninitialize the SDK. No fee is incurred for calling Init API.

Getting singleton

To use the voice feature, get the `ITMGContext` object first.

Sample code

```
import com.tencent.TMG.ITMGContext;
ITMGContext.getInstance(this);
```

Registering callback

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages.

Function prototype

Override this callback function in the constructor to process the parameters of the callback.

```
static public abstract class ITMGDelegate {
    public void OnEvent(ITMG_MAIN_EVENT_TYPE type, Intent data){}
}
```

Parameter	Type	Description
type	ITMGContext.ITMG_MAIN_EVENT_TYPE	Event type in the callback response
data	Intent message type	Callback message, i.e., event data

Sample code

Register the callback function to the SDK before room entry.

```
private ITMGContext.ITMGDelegate itmgDelegate = null;
itmgDelegate = new ITMGContext.ITMGDelegate() {
    @Override
    public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
        if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM == typ
        {
            // Analyze the returned data
            int nErrCode = data.getIntExtra("result" , -1);
            String strErrMsg = data.getStringExtra("error_info");
                }
        }
}
```

Function prototype

```
public abstract int SetTMGDelegate(ITMGDelegate delegate);
```

Parameter	Type	Description
delegate	ITMGDelegate	SDK callback function

Sample code

```
ITMGContext.GetInstance(this).SetTMGDelegate(itmgDelegate);
```

Initializing the SDK

This API is used to initialize the GME service. We recommend that you call it when initializing the application. No fees are incurred for calling this API.

For more information on how to get the `sdkAppID` parameter, see Activating Services.

`openID` uniquely identifies a user with the rules stipulated by you. It must be unique in the application and can only be in Int64 type.

Note:

The Init API must be called in the same thread with other APIs. It is recommended to call all APIs in the main thread.

Function prototype

```
public abstract int Init(String sdkAppId, String openId);
```

Parameter	Type	Description
sdkAppId	String	`AppId` from the GME console
openId	String	`OpenId` can only be in Int64 type, which is passed in after being converted to a string.

Returned Value	Description
QAVError.OK= 0	The SDK was initialized successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. We recommend that you delete it and then import it again.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is only a reminder but will not cause an initialization failure.

If this error is reported during integration, check the integrity and version of the SDK file as prompted.

If this error is returned after executable file export, ignore it and avoid displaying it on the UI.

Sample code

```
String sdkAppID = "14000*****";
String openID = "100";
int ret = ITMGContext.GetInstance(this).Init(sdkAppId, openId);
if(ret != 0){
                Log.e(TAG,"SDK initialization failed");
}
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. The `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.
You can refer to the `EnginePollHelper.java` file in the demo.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

Function prototype

```
public abstract int Poll();
```

Sample code

```
private Handler mhandler = new Handler();
private Runnable mRunnable = new Runnable() {
    @Override
    public void run() {
        if (s_pollEnabled) {
            if (ITMGContext.GetInstance(null) != null)
                ITMGContext.GetInstance(null).Poll();
        }
        mhandler.postDelayed(mRunnable, 33);
    }
};
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause.
If you need to pause the audio when switching to the background, you can call the `Pause` API in the listening code used to switch to the background, and call the `Resume` API in the listening event used to resume the foreground.

Function prototype

```
public abstract int Pause();
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

Function prototype

```
public abstract int Resume();
```

Uninitializing SDK

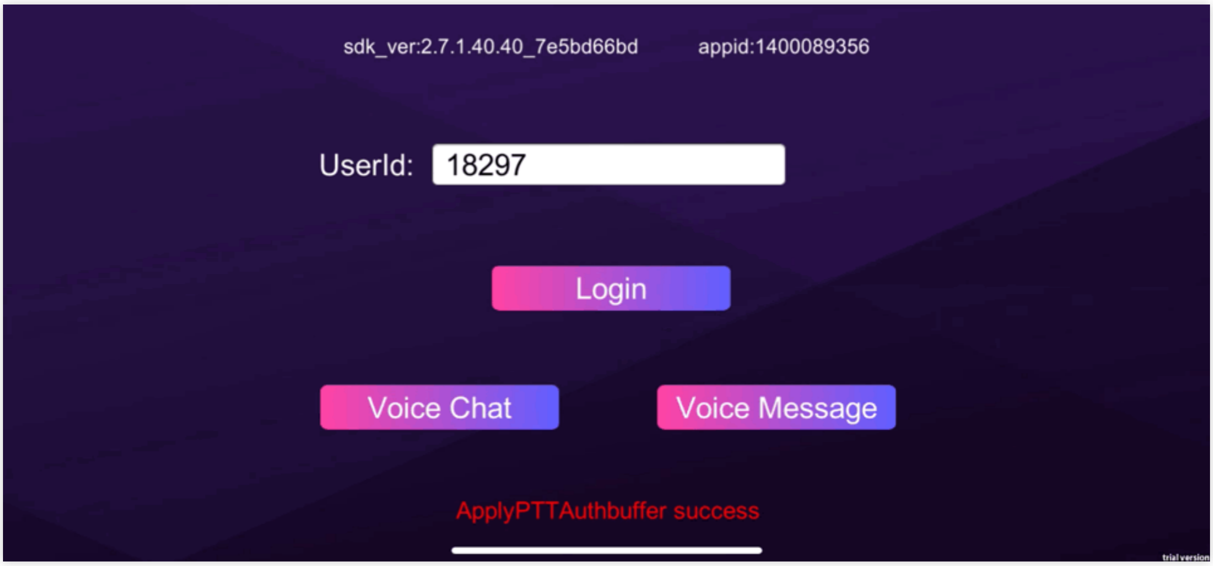
This API is used to uninitialize the SDK to make it uninitialized. **Switching accounts requires uninitialization**.

Function prototype

```
public abstract int Uninit();
```

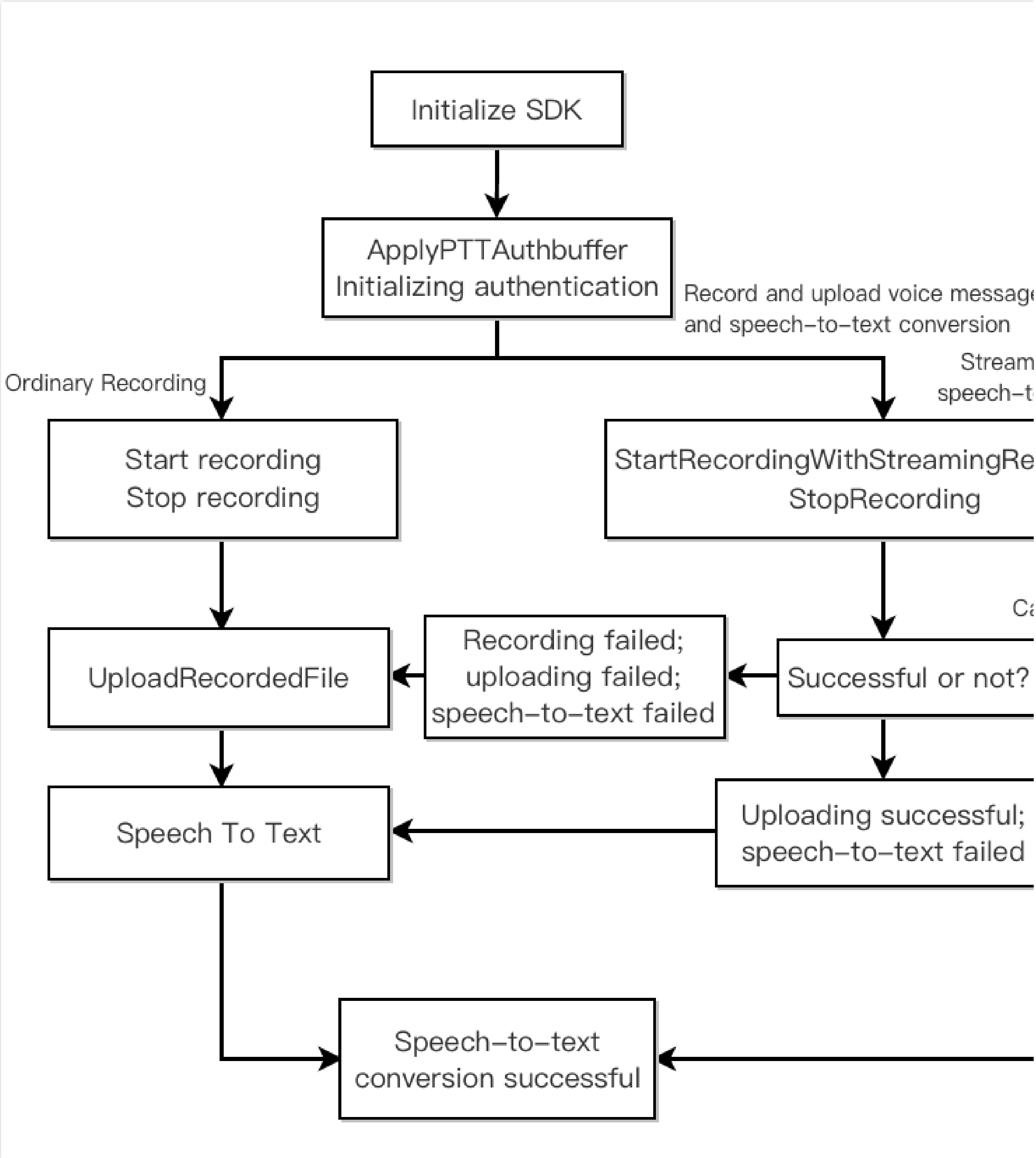
Voice Messaging and Speech-to-Text

Voice messaging refers to recording and sending a voice message. At the same time, the voice message can be converted to text and translated, as shown below:



Note:
We recommend that you use the streaming speech-to-text service.
You do not need to enter a voice chat room when using the voice messaging service.

Voice messaging and speech-to-text conversion flowchart



Integrating Voice Messaging and Speech-to-Text Service

Voice messaging and speech-to-text APIs

API	Description

ApplyPTTAuthbuffer	Initializes authentication.
SetMaxMessageLength	Specifies the maximum duration of a voice message.
StartRecording	Starts recording.
StartRecordingWithStreamingRecognition	Starts streaming recording.
PauseRecording	Pauses recording.
ResumeRecording	Resumes recording.
StopRecording	This API is used to stop audio recording.
CancelRecording	Cancels recording.
GetMicLevel	Gets the real-time mic volume level.
SetMicVolume	Sets the recording volume level.
GetMicVolume	Gets the recording volume level.
GetSpeakerLevel	Gets the real-time speaker volume level.
SetSpeakerVolume	Sets the playback volume level.
GetSpeakerVolume	Gets the playback volume level.
UploadRecordedFile	Uploads an audio file.
DownloadRecordedFile	Downloads an audio file.
PlayRecordedFile	Plays back an audio file.
StopPlayFile	Stops playing back an audio file.
GetFileSize	Gets the audio file size.
GetVoiceFileDuration	Gets the audio file duration.
SpeechToText	Converts speech to text.

Maximum recording duration

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, please call the `SetMaxMessageLength` API to set it after initialization.

Initializing the SDK

Before the initialization, the SDK is in the uninitialized status, and you need to initialize it through the `Init` API before you can use the voice chat and voice message services.
If you have any questions when using the service, see Speech-to-text Conversion.

Authentication information

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key. To get the authentication key for voice messaging and speech-to-text services, the room ID parameter must be set to null.

Function prototype

```
AuthBuffer public native byte[] genAuthBuffer(int sdkAppId, String roomId, String o
```

Parameter	Type	Description
appId	int	`AppId` from the Tencent Cloud console
roomId	string	Room ID, which must be set to null.
openId	string	User ID, which is the same as `openId` during initialization.
key	string	Permission key from the Tencent Cloud console.

Sample code

```
import com.tencent.av.sig.AuthBuffer;// Header file
byte[] authBuffer = AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(sdkAppI
```

Authentication initialization

Call authentication initialization after initializing the SDK. For more information on how to get the `authBuffer` , see `genAuthBuffer` (the voice chat authentication information API).

Function prototype

```
public abstract int ApplyPTTAuthbuffer(byte[] authBuffer);
```

Parameter	Type	Description
authBuffer	String	Authentication

Sample code

```
byte[] authBuffer =  AuthBuffer.getInstance().genAuthBuffer(Integer.parseInt(sdkApp
ITMGContext.GetInstance(this).GetPTT().ApplyPTTAuthbuffer(authBuffer);
```

Streaming Speech Recognition

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call `StopRecording`**. The callback will be returned after the recording is stopped.

Function prototype

```
public abstract int StartRecordingWithStreamingRecognition (String filePath);
public abstract int StartRecordingWithStreamingRecognition (String filePath,String
public abstract int StopRecording();
```

Parameter	Type	Description
filePath	String	Path of the stored audio file
speechLanguage	String	The language in which the audio file is to be converted into text. For parameters, see Language Parameter Reference List.
translateLanguage	String	The language into which the audio file is to be translated into text. For parameters, see Language Parameter Reference List. (This parameter is currently unavailable. Enter the same value as that of `speechLanguage`.)

Sample code

```
String temple = getActivity().getExternalFilesDir(null).getAbsolutePath() + "/test_
ITMGContext.GetInstance(getActivity()).GetPTT().StartRecordingWithStreamingRecognit
```

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen for callback messages in the callback function `onEvent`. Event messages are divided into:

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real time during the recording, which is equivalent to returning the recognized text while speaking.
The event message will be identified in the `OnEvent function` based on the actual needs. The delivered event message contains the following four parameters.

Parameter	Description
result	Return code indicating whether streaming speech-to-text conversion is successful
text	Text converted from speech
file_path	Local path of the stored recording file
file_id	Backend URL address of the recording file, which will be retained for 90 days. This field is fixed at http://gme-v2-.

Note:
The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNIT
            {
                /Callback for streaming voice message and speech-to-text
                Handler mainHander = new Handler(Looper.getMainLooper());
                mainHander.post(new Runnable() {
                    @Override
```

```
                    public void run() {
                if (nErrCode ==0) {

                    String textString = templeData.getStringExtra("text");
                    EditText _editText = (EditText) root.findViewById(R.id.edit_aud
                    _editText.setText(textString);

                    recordfilePath = templeData.getStringExtra("file_path");
                    mEditTextfiletoupload.setText(recordfilePath);

                    donwLoadUrlPath = templeData.getStringExtra("file_id");
                    mEditTextDownloadurl.setText(donwLoadUrlPath);

                    Log.e(TARGET, "STREAMINGRECOGNITION" + "nErrCode=" + nErrCode +
                }
                else
                {

                    Toast.makeText(getActivity(), String.format("Streaming speech-t
                    Log.e(TARGET, "Streaming speech-to-text conversion failed. Erro
                }
                    }
                });
            }
}
```

Voice Message Recording

The recording process is as follows: start recording > stop recording > return recording callback > start the next recording.

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

Function prototype

```
public abstract int SetMaxMessageLength(int msTime);
```

Parameter	Type	Description
msTime	int	Audio duration in ms. Value range: 1000 < `msTime` < 58000.

Sample code

```
ITMGContext.GetInstance(this).GetPTT().SetMaxMessageLength(msTime);
```

Starting recording

This API is used to start recording. The recording file must be uploaded first before you can perform operations such as speech-to-text conversion. **To stop recording, call `StopRecording`** .

Function prototype

```
public abstract int StartRecording(String filePath);
```

Parameter	Type	Description
filePath	String	Path of the stored audio file

Sample code

```
ITMGContext.GetInstance(this).GetPTT().StartRecording(filePath);
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

Function prototype

```
public abstract int StopRecording();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().StopRecording();
```

Callback for recording start

A callback will be executed through a delegate function to pass a message when recording is completed.
To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.
The callback function `OnEvent` will be called after recording is started. The event message `ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE` will be returned, which will be identified in the `OnEvent`

function.

The passed parameter includes `result` and `file_path` .

Error codes

Error Code	Reasons	Suggested Solution
4097	Empty parameters.	Check whether the API parameters in the code are correct.
4098	An initialization error occurred.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
4099	Recording is in progress.	Make sure that the SDK recording feature is used at the right time.
4100	No audio data is captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	The mic permission is required for using the SDK. To add the permission, see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE =
            {
                    // Callback for recording start
                     if (nErrCode ==0)
                        {
                            recordfilePath = templeData.getStringExtra("file_path")
                            mEditTextfiletoupload.setText(recordfilePath);
                        }
            }
}
```

Pausing recording

This API is used to pause recording. If you want to resume recording, call the `ResumeRecording` API.

Function prototype

```
public abstract int PauseRecording();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().PauseRecording();
```

Resuming recording

This API is used to resume recording.

Function prototype

```
public abstract  int ResumeRecording();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().ResumeRecording();
```

Canceling recording

This API is used to cancel recording. There is no callback after cancellation.

Function prototype

```
public abstract int CancelRecording();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().CancelRecording();
```

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. An int-type value will be returned. Value range: 0-200.

This API is different from the voice chat API and is in `ITMGPTT.java`.

Function prototype

```
public abstract int GetMicLevel();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetMicLevel();
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

This API is different from the voice chat API and is in `ITMGPTT.java` .

Function prototype

```
public abstract int SetMicVolume(int volume);
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().SetMicVolume(100);
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. An int-type value will be returned. Value range: 0-200.

This API is different from the voice chat API and is in `ITMGPTT.java` .

Function prototype

```
public abstract int GetMicVolume();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetMicVolume();
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. An int-type value will be returned. Value range: 0-200.

This API is different from the voice chat API and is in `ITMGPTT.java` .

Function prototype

```
public abstract int GetSpeakerLevel();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetSpeakerLevel();
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice message. Value range: 0-200.

This API is different from the voice chat API and is in `ITMGPTT.java`.

Function prototype

```
public abstract int SetSpeakerVolume(int volume);
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().SetSpeakerVolume(100);
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice message. An int-type value will be returned. Value range: 0-200.

This API is different from the voice chat API and is in `ITMGPTT.java`.

Function prototype

```
public abstract int GetSpeakerVolume();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetSpeakerVolume();
```

Voice Message Playback

Playing back audio

This API is used to play back audio.

Function prototype

```
public abstract int PlayRecordedFile(String filePath);public abstract int PlayRecor
```

Parameter	Type	Description

downloadFilePath	String	Local audio file path
voicetype	int	Voice changing type. For more information, see Voice Changing.

Error codes

Error Code	Reasons	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
ITMGContext.GetInstance(this).GetPTT().PlayRecordedFile(downloadFilePath);
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameter includes `result` and `file_path`.

Error codes

Error Code	Reasons	Suggested Solution
20481	An initialization error occurred.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Empty parameters.	Check whether the API parameters in the code are correct.
20484	An internal error occurred.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if(ITMGContext.ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE== type)
```

```
                    {
                // Callback for audio playback
            }
}
```

Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

Function prototype

```
public abstract int StopPlayFile();
```

Sample code

```
ITMGContext.GetInstance(this).GetPTT().StopPlayFile();
```

Getting audio file size

This API is used to get the size of an audio file.

Function prototype

```
public abstract int GetFileSize(String filePath);
```

Parameter	Type	Description
filePath	String	Path of the audio file, which is a local path.

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetFileSize(path);
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

Function prototype

```
public abstract int GetVoiceFileDuration(String filePath);
```

Parameter	Type	Description

filePath	String	Path of the audio file, which is a local path.

Sample code

```
ITMGContext.GetInstance(this).GetPTT().GetVoiceFileDuration(path);
```

Voice Message Upload and Download

Uploading an audio file

This API is used to upload an audio file.

Function prototype

```
public abstract int UploadRecordedFile(String filePath);
```

Parameter	Type	Description
filePath	String	Path of the uploaded audio file, which is a local path.

Sample code

```
ITMGContext.GetInstance(this).GetPTT().UploadRecordedFile(filePath);
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameters include `result` , `file_path` , and `file_id` .

Error codes

Error Code	Reasons	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.
8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.

8195	Network error	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device network can normally access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device network can normally access the internet.
8200	`appinfo` is not set.	Check whether the `apply` API is called or whether the input parameter is not specified or null.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if(ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE==
            {
               // Callback for audio file upload completion
            }
}
```

Downloading the audio file

This API is used to download an audio file.

Function prototype

```
public abstract int DownloadRecordedFile(String fileID, String filePath);
```

Parameter	Type	Description
fileID	String	File URL
downloadFilePath	String	Local path of the saved file

Sample code

```
ITMGContext.GetInstance(this).GetPTT().DownloadRecordedFile(url,path);
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function. The passed parameters include `result` 、 `file_path` and `file_id` .

Error codes

Error Code	Reasons	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
12291	A network storage system exception occurred.	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	A server file system error occurred.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed while getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.
12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.
12297	`appinfo` is not set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if(ITMGContext.ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE== type)
        {
            //Download succeeded
    }
}
```

Speech-to-Text Service

Converting audio file to text

This API is used to convert a specified audio file to text.

Function prototype

```
public abstract int SpeechToText(String fileID);
```

Parameter	Type	Description
fileID	String	Audio file URL

Sample code

```
ITMGContext.GetInstance(this).GetPTT().SpeechToText(fileID);
```

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Function prototype

```
public abstract int SpeechToText(String fileID, String speechLanguage,String transl
```

Parameter	Type	Description
fileID	String	URL of the audio file, which will be retained on the server for 90 days.
speechLanguage	String	The language in which the audio file is to be converted into text. For parameters, see Language Parameter Reference List.
translatelanguage	String	The language into which the audio file is to be translated into text. For parameters, see Language Parameter Reference List. (This parameter is currently unavailable. Enter the same value as that of `speechLanguage` .)

Sample code

```
ITMGContext.GetInstance(this).GetPTT().SpeechToText(fileID,"cmn-Hans-CN","cmn-Hans-
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result`, `file_path` and `text` (recognized text).

Error codes

Error Code	Reasons	Suggested Solution
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network connection failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	`appinfo` is not set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.
32784	The speech-to-text conversion parameter is incorrect.	Check whether the API parameter `fileid` in the code is empty.
32785	A speech-to-text translation error occurred.	An error occurred in the voice messaging and speech-to-text feature on the backend. Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.

Sample code

```
public void OnEvent(ITMGContext.ITMG_MAIN_EVENT_TYPE type, Intent data) {
    if(ITMGContext.ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE == type)
            {
            //Recognized audio file successfully
      }
}
```


Advanced APIs

Getting version number

This API is used to get the SDK version number for analysis.

Function prototype

```
public abstract String GetSDKVersion();
```

Sample code

```
ITMGContext.GetInstance(this).GetSDKVersion();
```

Checking mic permission

This API is used to return the mic permission status.

Function prototype

```
public abstract ITMG_RECORD_PERMISSION  CheckMicPermission();
```

Parameter description

Parameter	Value	Description
ITMG_PERMISSION_GRANTED	0	The mic permission is granted.
ITMG_PERMISSION_Denied	1	Mic disabled.
ITMG_PERMISSION_NotDetermined	2	No authorization box has been popped up to request the permission.
ITMG_PERMISSION_ERROR	3	An error occurred while calling the API.

Sample code

```
ITMGContext.GetInstance(this).CheckMicPermission();
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

Function prototype

```
public abstract int SetLogLevel(int levelWrite, int levelPrint);
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO` .
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: `TMG_LOG_LEVEL_ERROR` .

ITMG_LOG_LEVEL

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContext.GetInstance(this).SetLogLevel(TMG_LOG_LEVEL_INFO,TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path, which is /sdcard/Android/data/xxx.xxx.xxx/files by default.

Function prototype

```
public abstract int SetLogPath(String logDir);
```

Parameter	Type	Description

logDir	String	Path

Sample code

```
ITMGContext.GetInstance(this).SetLogPath(path);
```

Callback Messages

Message list

Message	Description
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	PTT recording is completed.
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	PTT upload is completed.
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	PTT download is completed.
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	PTT playback is completed.
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Speech-to-text conversion is completed.

Data list

Message	Data	Example
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	result; file_path	{"file_path":
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	result; file_path;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	result; file_path;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	result; file_path	{"file_path":
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	result; text;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	result; file_path; text;file_id	{"file_id":"",'

Project Export

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document mainly describes the notes on exporting the Android project so that the Android developers can easily debug and integrate the APIs for Game Multimedia Engine (GME).

Project Export

The GME SDK provides lib files for v7a, v8a, x86, and x86_64 by default. Please delete unnecessary files as needed.

Warning

If the .so file of the corresponding architecture is missing during the running of the Android system device, the system will crash.

Configuring Permissions

Required permissions

Add the following permissions in the `AndroidManifest.xml` file of the project:

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

Adding permissions as needed

Add the following permissions in the `AndroidManifest.xml` file of the project as needed:

Read/Write

Bluetooth permission

The read/write permission is not required. Determine whether to add it according to the following rules:

If you use the default log path (`/sdcard/Android/data/xxx.xxx.xxx/files`), it means that you do not call `SetLogPath` and do not need the `WRITE_EXTERNAL_STORAGE` permission.

If you call the `SetLogPath` API to set the log path to an external storage device, and the storage path of the voice message recording is an external storage device, you need to apply for the `WRITE_EXTERNAL_STORAGE` permission to the user and get the user's approval.

You don't need to add this permission for devices on Android 6 or later.


```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

Add the Bluetooth permission according to the following rules:

If `targetSDKVersion` in the project is v30 or earlier:

```
<uses-permission android:name="android.permission.BLUETOOTH"/>
```

If `targetSDKVersion` in the project is v31 or later and GME is earlier than v2.9.6:

```
<uses-permission android:name="android.permission.BLUETOOTH" android:maxSdkVersion=
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
```

App obfuscation

If you want to obfuscate the code, configure the following:

```
-dontwarn com.tencent.**
-keep class com.tencent.** { *;}
-keepclassmembers class com.tencent.**{*;}
```

Note that after **v2.9.0**, obfuscation is required with the following configurations.

```
-dontwarn com.gme.**
-keep class com.gme.** { *;}
-keepclassmembers class com.gme.**{*;}
```

Advanced Android Configuration

According to Behavior changes: all apps for Android 9 on the Android Developers platform, Android 9 limits background apps' access to user inputs and sensor data, that is, apps running in the background cannot access the mic or camera.

If Android 9 users need to continue capturing audio or video after locking the screen, a service can be initiated before the screen is locked or the app is brought to the background and terminated before the screen is unlocked or the app is brought to the foreground.

Android Project Export FAQs

Project problems occurred during or after the export of the executable files:

SDK for macOS
Project Configuration

최종 업데이트 날짜: : 2024-01-18 15:13:51

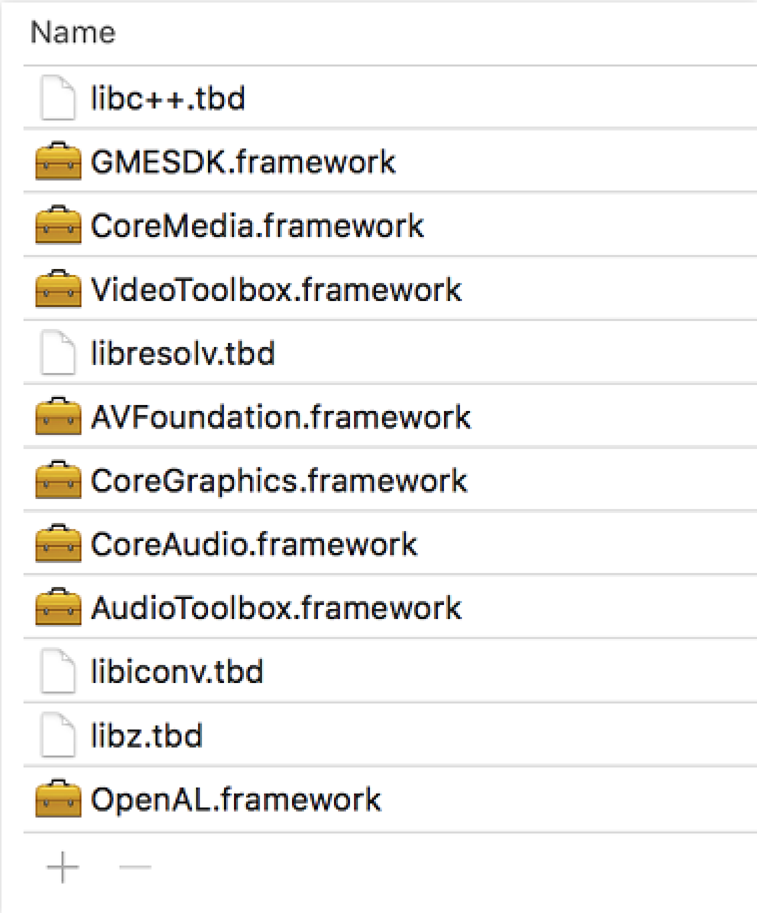
This document describes how to configure a macOS project for the GME APIs for macOS.

SDK Preparations

1. Download the applicable demo and SDK. For more information, please see Download Guide.
2. Decompress the obtained SDK resources.
3. The extracted `GMESDK.framework` is the resource related to GME.

Configuration Guide

Add the following dependent libraries to `Link Binary With Libraries` in Xcode as needed and configure `Framework Search Paths` to point to the directory where the SDK is located as shown below:


Name
libc++.tbd
GMESDK.framework
CoreMedia.framework
VideoToolbox.framework
libresolv.tbd
AVFoundation.framework
CoreGraphics.framework
CoreAudio.framework
AudioToolbox.framework
libiconv.tbd
libz.tbd
OpenAL.framework

Voice Chat API

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document describes how to integrate with Game Multimedia Engine (GME) on macOS.

Note:

This document applies to GME SDK version 2.9.

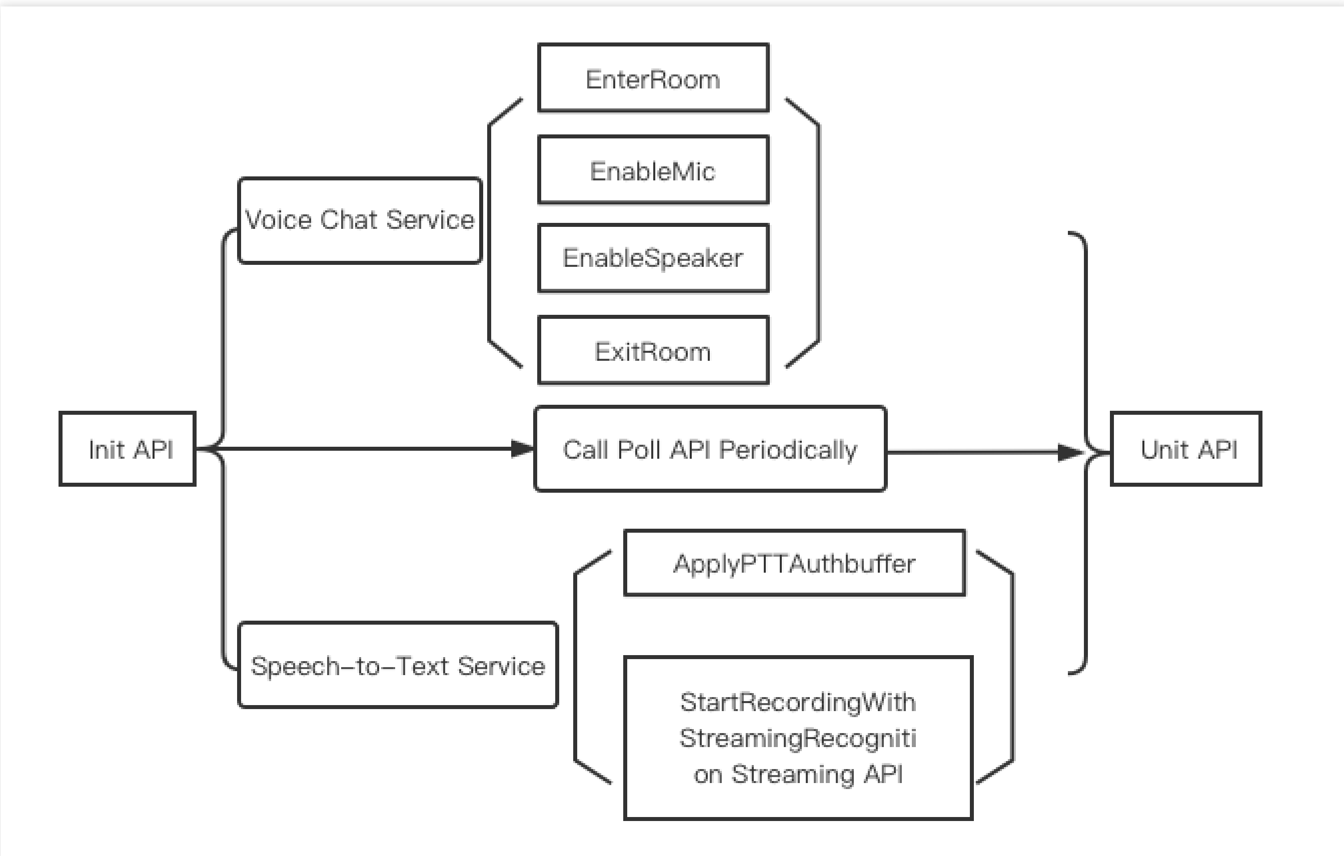
Considerations

GME provides two services: voice chat service and voice message and speech-to-text service, both of which rely on key APIs such as Init and Poll.

Note on Init API

If you need to use voice chat and voice message services at the same time, **you only need to call** `Init` **API once**.

Billing will not start after initialization. After you call Entering a voice chat room to enter the room successfully, the billing will start.



Directions

1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Entering a voice chat room, API: EnterRoom
4. Enabling the microphone, API: EnableMic
5. Enabling the speaker, API: EnableSpeaker
6. Exiting a voice room, API: ExitRoom
7. Uninitializing GME, API: UnInit

Important notes

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error code, please see Error Codes.

APIs

```
@class ITMGRoom;//Room APIs
@class ITMGAudioCtrl;//Audio APIs
@class ITMGAudioEffectCtrl;//Sound effect, accompaniment APIs
```

Key APIs

Before the initialization, the SDK is in the uninitialized status, and **you need to initialize it through the `Init` API** before you can use the voice chat and speech-to-text services.

You need to call the `Init` API before calling any APIs of GME.

If you have any questions when using the service, please see General FAQs.

API	Description
InitEngine	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME

Getting singleton

To use the voice feature, get the `ITMGContext` object first.

Function prototype

```
ITMGContext ITMGDelegate <NSObject>
```

Sample code

```
ITMGContext* _context = [ITMGContext GetInstance];
_context.TMGDelegate =self;
```

Message delivery

The API callback messages is processed in `OnEvent` . For the message type, please see `ITMG_MAIN_EVENT_TYPE` . The message content is a dictionary for parsing the API callback contents.

Function prototype

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data;
```

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
        NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
        switch (eventType) {
            //Identify `eventType`
            }
    }
```

Initializing SDK

This API is used to initialize the GME service. It is recommended to call it when initializing the application. No fee is incurred for calling this API.

For more information on how to get the `sdkAppID` parameter, see Voice Service Activation Guide.

The openID uniquely identifies a user with the rules stipulated by the application developer and unique in the application (currently, only INT64 is supported).

Note:

The SDK must be initialized before a user can enter a voice chat room.

The Init API must be called in the same thread with other APIs. It is recommended to call all APIs in the main thread.

Function prototype

```
-(int)InitEngine:(NSString*)sdkAppID openID:(NSString*)openID;
```

Parameter	Type	Description
sdkAppId	String	`AppId` provided by the GME service from the Tencent Cloud console
OpenId	String	`OpenId` can only be in Int64 type, which is passed after being converted to a string.

Returned values

Returned Value	Description
QAV_OK= 0	Initialized SDK successfully.
QAV_ERR_SDK_NOT_FULL_UPDATE= 7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.
If this error is reported during integration, please check the integrity and version of the SDK file as prompted.
If this error is returned after executable file export, please ignore it and try to avoid displaying it in the UI.

Sample code

```
_openId = _userIdText.text;
_appId = _appIdText.text;
[[ITMGContext GetInstance] InitEngine:SDKAPPID openID:_openId];
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. The `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.
Refer to the EnginePollHelper.m file in Demo.

Calling the `Poll` API periodically

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

Function prototype

```
-(void)Poll;
```

Sample code

```
[[ITMGContext GetInstance] Poll];
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause.
If you need to pause the audio when switching to the background, you can call the `Pause` API in the listening code used to switch to the background, and call the `Resume` API in the listening event used to resume the foreground.

Function prototype

```
-(QAVResult)Pause;
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

Function prototype

```
-(QAVResult)Resume;
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **Switching accounts requires uninitialization**.

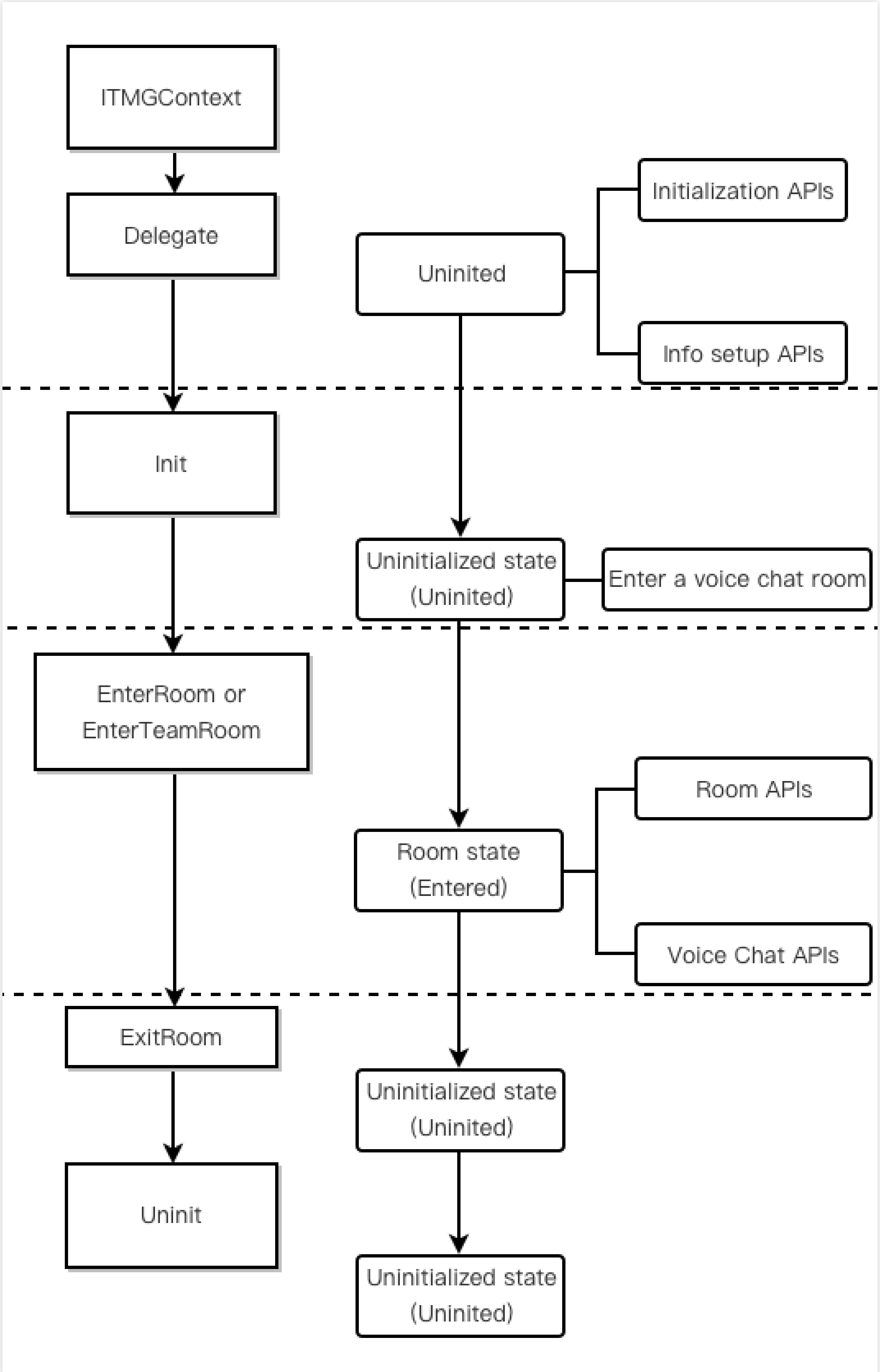
Function prototype

```
-(int)Uninit;
```

Sample code

```
[[ITMGContext GetInstance] Uninit];
```

Voice chat flowchart



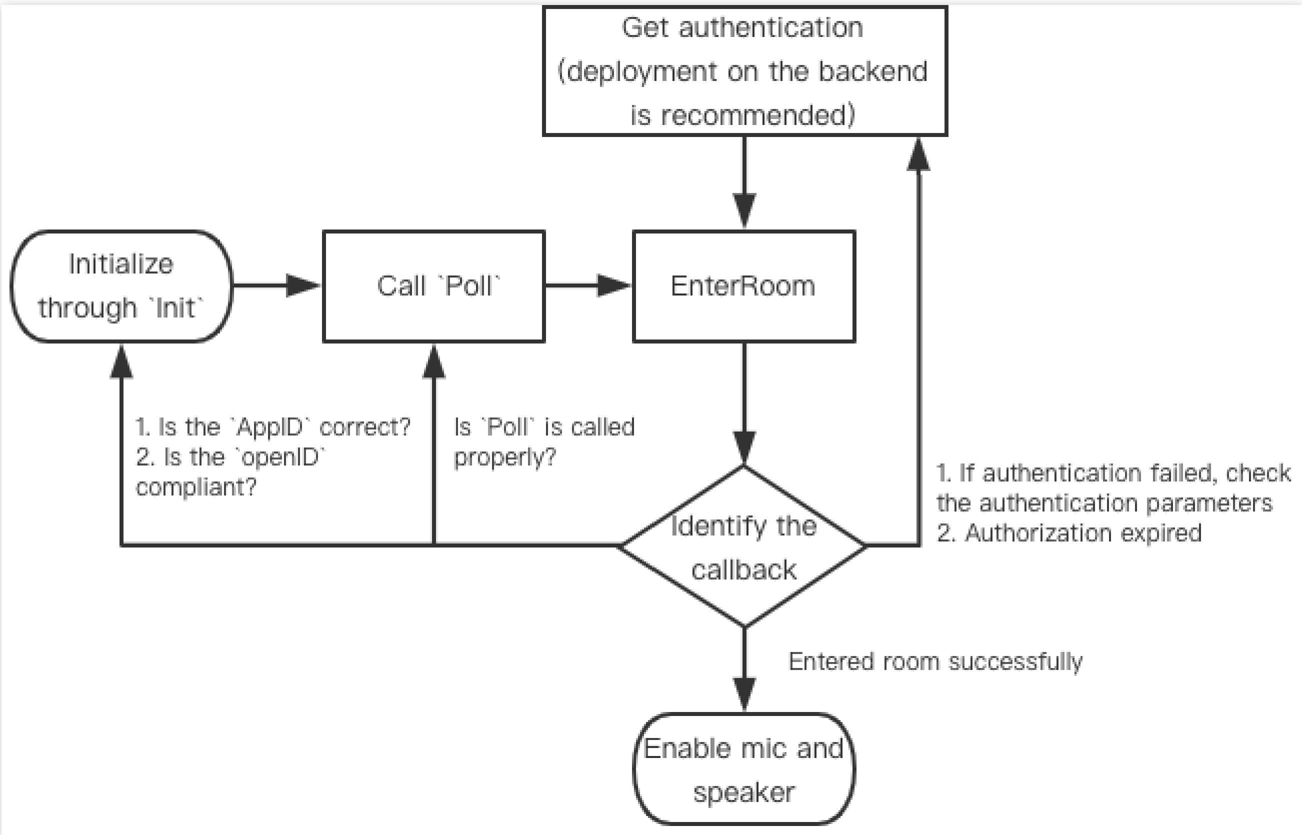
Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, please see FAQs About Voice Chat.

API	Description
GenAuthBuffer	Initializes authentication
EnterRoom	Enters room
IsRoomEntered	Indicates whether room entry is successful
ExitRoom	Exits room
ChangeRoomType	Modifies user's room audio type
GetRoomType	Gets user's room audio type

Voice chat room call flowchart



Entered the room successfully

If the room entry callback result is 0, the room entry is successful. The returned value of 0 from the `EnterRoom` API does not necessarily mean that the room entry is successful.

Authentication information

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.

Function prototype

```
@interface QAVAuthBuffer : NSObject
+ (NSData*) GenAuthBuffer:(unsigned int)appId roomId:(NSString*)roomId openID:(NSSt
+ @end
```

Parameter	Type	Description
appId	int	`AppId` from the Tencent Cloud console.
roomId	NSString	Room ID, which can contain up to 127 characters.
openID	NSString	User ID, which is the same as `openID` during initialization.
key	NSString	Permission key from the Tencent Cloud console.

Sample code

```
#import "GMESDK/QAVAuthBuffer.h"
NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:SDKAPPID3RD.intValue roomId:_room
```

Entering a room

When a user enters a room with the generated authentication information, the `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` message will be received as a callback. Mic and speaker are not enabled by default after room entry. The returned value of `AV_OK` indicates a success.
The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will become smooth sound quality. Only when a member in the room calls the `ChangeRoomType` API, the audio type of the room will be changed.
For more information on how to choose a room audio type, please see Sound Quality Selection.

Function prototype

```
-(int)EnterRoom:(NSString*) roomId roomType:(int)roomType authBuffer:(NSData*)authB
```

Parameter	Type	Description
roomId	NSString	Room ID, which can contain up to 127 characters
roomType	int	Room audio type
authBuffer	NSData	Authentication key

Sample code

```
[[ITMGContext GetInstance] EnterRoom:_roomId roomType:_roomType authBuffer:authBuff
```

Callback for room entry

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function for callback and processing. A successful callback means that the room entry is successful, and the billing starts.

Billing references

Purchase Guide

Billing FAQs

Will the billing continues if the client goes offline when using Voice Chat?

Sample code

Sample code for processing the callback, including room entry and network disconnection events.

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
            int result = ((NSNumber*)[data objectForKey:@"result"]).intValue;
            NSString* error_info = [data objectForKey:@"error_info"];
                //Receive the event of successful room entry
        }
            break;
    }
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnected callback prompt `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT`. At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START`. When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS`.

Error codes

Error Code Value	Cause and Suggested Solution
7006	Authentication failed: The `AppID` does not exist or is incorrect. An error occurred while authenticating the `authbuff`. Authentication expired. The `OpenId` does not meet the specification.
7007	Already in another room.
1001	The user was already in the process of entering a room but repeated this operation. It is recommended not to call the room entering API until the room entry callback is returned.
1003	The user was already in the room and called the room entering API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally.

Exiting the room

This API is called to exit the current room. It is an async API. The returned value of `AV_OK` indicates a successful async delivery.

Note:

If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API during API call; instead, you can directly call the API.

Function prototype

```
-(int)ExitRoom
```

Sample code

```
[[ITMGContext GetInstance] ExitRoom];
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM` .

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
        {
            // Receive the event of successful room exit
        }
            break;
    }
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"","result":0}

Determining whether user has entered room

This API is used to determine whether the user has entered a room. A bool-type value will be returned. The call is invalid during the process of room entry.

Function prototype

```
-(BOOL)IsRoomEntered;
```

Sample code

```
[[ITMGContext GetInstance] IsRoomEntered];
```

Switching room

User can call this API to quickly switch the voice chat room after entering the room. After the room is switched, the device is not reset, that is, if the microphone is already enabled in this room, the microphone will keep enabled after the room is switched.

The callback for quickly switching rooms is `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM` , and the fields are `error_info` and `result` .

API prototype

```
-(int) SwitchRoom:(NSString *)roomID  authBuffer:(NSData*)authBuffer;
```

Type descriptions

Parameter	Type	Description
targetRoomID	NSString *	ID of the room to enter
authBuffer	NSData*	Generates a new authentication with the ID of the room to enter

Callback sample code

```
- (IBAction)swichRoom:(id)sender {
    NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:_appId.intValue roomID:_roomI
    [[[ITMGContext GetInstance]GetRoom]SwitchRoom:_roomIdText.text authBuffer:authB
}

-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSString* log =[NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventType
    [self showLog:log];
    NSLog(@"====%@====",log);
    switch (eventType) {
    case ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM:
        {
            int result = ((NSNumber*)[data objectForKey:@"result"]).intValue;
            NSString* log = nil;
            if (result == QAV_OK) {
```



```
                log = [NSString stringWithFormat:@"switch room success."];
            } else {
                log = [NSString stringWithFormat:@"switch room failed."];
            }
            [self showLog:log];
            break;
        }
    }
}
```

Cross-room mic connection

Call this API to connect the microphones across rooms after entering the room. And the local user can communicate with the target OpenID user in the target room.

Example

User a is in room A, user b is in room B, and user a can talk with b through the cross-room API. When user c in room A speaks, users b and d in room B cannot hear. User c in room A can hear only the voice in room A and the voice of user b in room B but not other users in room B.

API prototype

```
-(int) StartRoomSharing:(NSString *)targetRoomID targetOpenID:(NSString *)targetOpe

-(int) StopRoomSharing;
```

Type descriptions

Parameter	Type	Description
targetRoomID	NSString *	ID of the room to connect mic
targetOpenID	NSString *	The target OpenID to connect mic
authBuffer	NSData*	Reserved flag. You just need to enter NULL.

Sample code

```
- (IBAction)shareRoom:(id)sender {
    if(_shareRoomSwitch.isOn){
        [[[ITMGContext GetInstance]GetRoom]StartRoomSharing:_shareRoomID.text targe
    }else{
```

```
            [[[ITMGContext GetInstance]GetRoom]StopRoomSharing];
        }
    }
}
```

Notifications of member room entry and speaking status

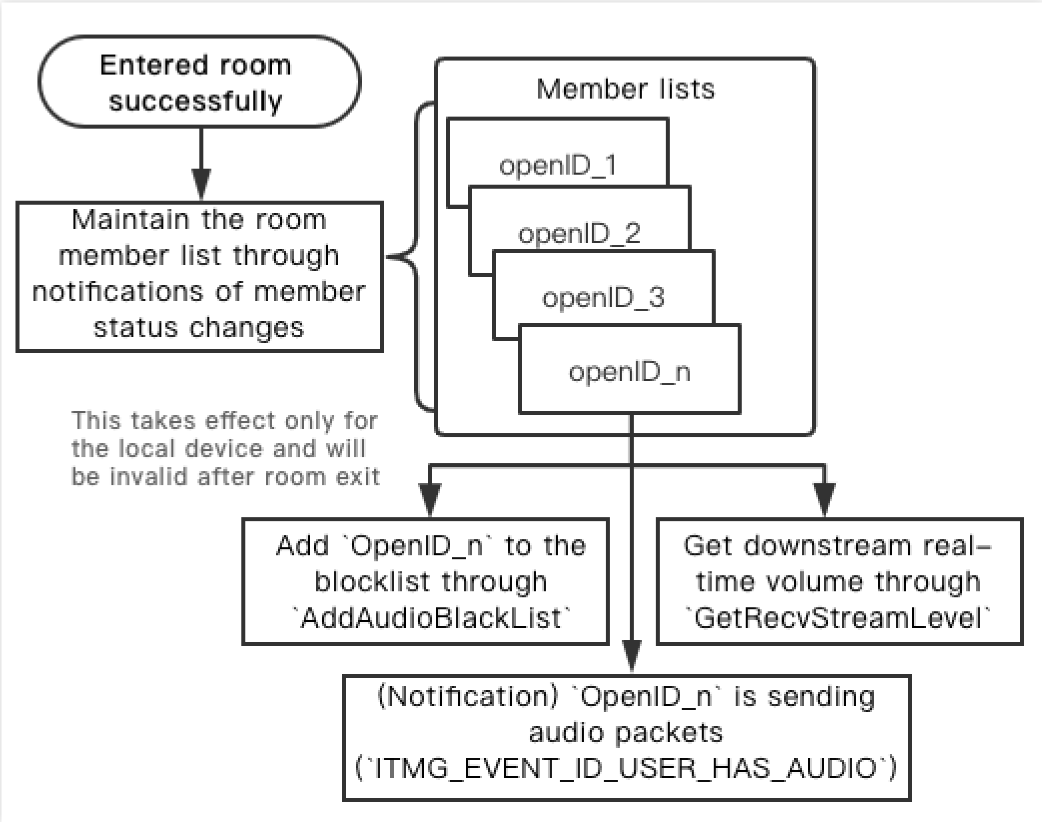
This API is used to obtain the user speaking in the room and display it in the UI, and to send a notification when someone entering or exiting the room.

Notification for this event will be sent only when the status changes. To get member status in real time, cache the notification when it is received at a higher layer. The event message is `ITMG_MAIN_EVNET_TYPE_USER_UPDATE` , where the data contains `event_id` and `user_list` . The event message will be identified in the `OnEvent` function.

Notifications for audio events are subject to a threshold, and a notification will be sent only when this threshold is exceeded. The notification "A member has stopped sending audio packets" will be sent if no audio packets are received in more than two seconds. This event only returns the member speaking event, but not the specific volume level. If you need the specific volume levels of members in the room, use the `GetVolumeById` API.

event_id	Description	Maintenance
ITMG_EVENT_ID_USER_ENTER	A member enters the room	Member list
ITMG_EVENT_ID_USER_EXIT	A member exits the room	Member list
ITMG_EVENT_ID_USER_HAS_AUDIO	A member sends audio packets. This event can be used to determine whether a user is speaking and display the voiceprint effect. It can be called together with getRecvStreamLevel.	Chat member list
ITMG_EVENT_ID_USER_NO_AUDIO	A member stops sending audio packets	Chat member list

Room member maintenance flowchart



Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    ITMG_EVENT_ID_USER_UPDATE event_id=((NSNumber*)[data objectForKey:@"event_id"])
    NSMutableArray* uses = [NSMutableArray arrayWithArray: [data objectForKey:@"use
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_USER_UPDATE:
        {
        // Process
        //Parse the parameter to get `event_id` and `user_list`
            switch (eventID)
             {
             case ITMG_EVENT_ID_USER_ENTER:
                  // A member enters the room
                  break;
             case ITMG_EVENT_ID_USER_EXIT:
                  // A member exits the room
                break;
            case ITMG_EVENT_ID_USER_HAS_AUDIO:
                // A member sends audio packets
                break;
            case ITMG_EVENT_ID_USER_NO_AUDIO:
                // A member stops sending audio packets
```

```
                    break;
                }
            break;
            }
        }
}
```

Data details

Message	Data	Sample
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	event_id; user_list	{"event_id":0,"user_list":""}

Muting in a room

This API is used to add an ID to the audio data blacklist. This operation blocks audio from someone and only applies to the local device. A returned value of `0` indicates the call is successful. Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B;

If B blocks neither A nor C, B can hear both of them;

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

Function prototype

```
ITMGContext GetAudioCtrl -(QAVResult)AddAudioBlackList:(NSString*)openID;
```

Parameter	Type	Description
openId	NSString	ID to be blocked openid

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ] AddAudioBlackList[id]];
```

Unmuting

This API is used to remove an ID from the audio data blocklist. A returned value of `0` indicates the call is successful.

Function prototype

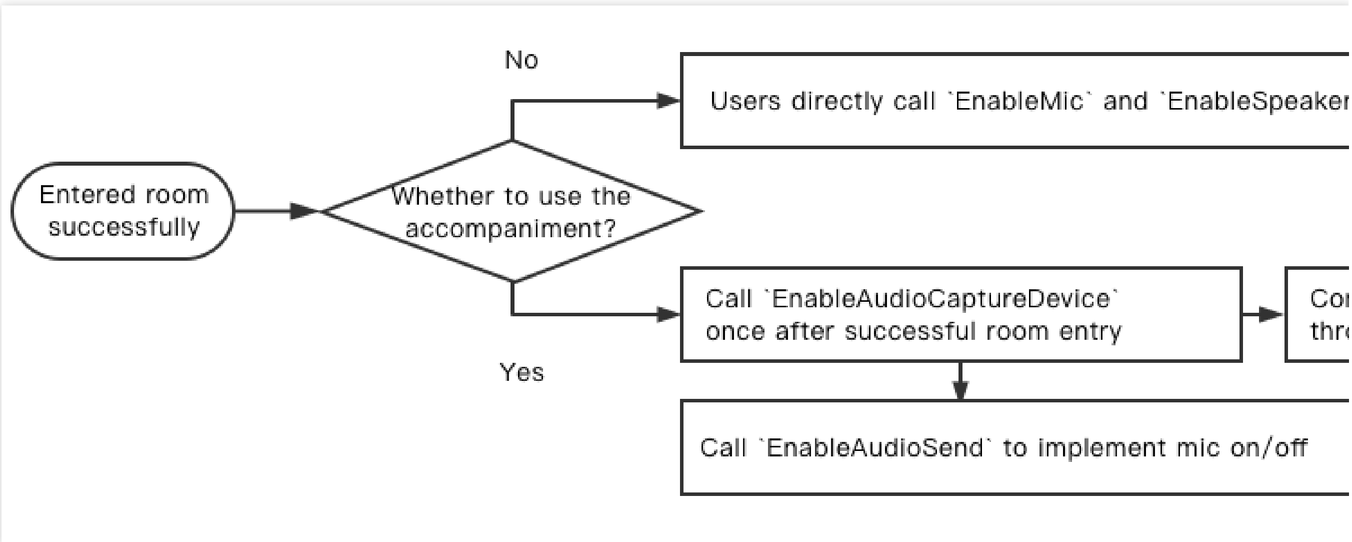
```
-(QAVResult)RemoveAudioBlackList:(NSString*)openID;
```

Parameter	Type	Description
openId	NSString	ID to be unblocked openid

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ] RemoveAudioBlackList[openId]];
```

Voice Chat Audio APIs



Notes on voice chat audio connection

The voice chat APIs can only be called after SDK initialization and room entry.

When Enable/Disable Mic/Speaker is clicked on the UI, the following practices are recommended:

For most game applications, it is recommended to call the `EnableMic` and `EnableSpeaker` APIs, which is equivalent to calling the `EnableAudioCaptureDevice/EnableAudioSend` and `EnableAudioPlayDevice/EnableAudioRecv` APIs.

For other mobile applications (such as social networking applications), enabling/disabling a capturing device will restart both capturing and playback devices. If the application is playing back background music, it will also be interrupted. Playback will not be interrupted if the mic is enabled/disabled through control of upstreaming/downstreaming. **Calling method: call `EnableAudioCaptureDevice(true)` and `EnableAudioPlayDevice(true)` once after room entry, and call `EnableAudioSend/Recv` to send/receive audio streams when Enable/Disable Mic is clicked**.

For more information on how to release only a capturing or playback device, please see the `EnableAudioCaptureDevice` and `EnableAudioPlayDevice` .

Call the `pause` API to pause the audio engine and call the `resume` API to resume the audio engine.

Voice chat audio APIs

API	Description
EnableMic	Enables/disables mic
GetMicState	Gets mic status
EnableAudioCaptureDevice	Enables/disables capturing device
IsAudioCaptureDeviceEnabled	Gets capturing device status
EnableAudioSend	Enables/disables audio upstreaming
IsAudioSendEnabled	Gets audio upstreaming status
GetMicLevel	Gets real-time mic volume
GetSendStreamLevel	Gets real-time audio upstreaming volume
SetMicVolume	Sets mic volume
GetMicVolume	Gets mic volume
EnableSpeaker	Enables/disables speaker
GetSpeakerState	Gets speaker status
EnableAudioPlayDevice	Enables/disables playback device
IsAudioPlayDeviceEnabled	Gets playback device status
EnableAudioRecv	Enables/disables audio downstreaming
IsAudioRecvEnabled	Gets audio downstreaming status
GetSpeakerLevel	Gets real-time speaker volume
GetRecvStreamLevel	Gets real-time downstreaming audio levels of other members in room

SetSpeakerVolume	Sets speaker volume
GetSpeakerVolume	Gets speaker volume
EnableLoopBack	Enables/disables in-ear monitoring

Voice Chat Capturing APIs

Enabling or disabling the microphone

This API is used to enable/disable the mic. Mic and speaker are not enabled by default after room entry.

If accompaniment is used, please call this API as instructed in Accompaniment in Voice Chat.

EnableMic = EnableAudioCaptureDevice + EnableAudioSend

Function prototype

```
-(QAVResult)EnableMic:(BOOL)enable;
```

Parameter	Type	Description
isEnabled	boolean	To enable the mic, set this parameter to `YES` ; otherwise, set it to `NO` .

Sample code

```
// Enable mic
[[[ITMGContext GetInstance] GetAudioCtrl] EnableMic:YES];
```

Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

Function prototype

```
-(int)GetMicState;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetMicState];
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.

This API can only be called after room entry. The device will be disabled automatically after room exit.

Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

Function prototype

```
-(QAVResult)EnableAudioCaptureDevice:(BOOL)enabled;
```

Parameter	Type	Description
enabled	BOOL	To enable the capturing device, set this parameter to `YES` , otherwise set it to `NO` .

Sample code

```
// Enable capturing device
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioCaptureDevice:enabled];
```

Getting the capturing device status

This API is used to get the status of a capturing device.

Function prototype

```
-(BOOL)IsAudioCaptureDeviceEnabled;
```

Sample code

```
BOOL IsAudioCaptureDevice = [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioCaptur
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain mute. For more information on how to enable/disable the capturing device, please see the `EnableAudioCaptureDevice` API.

Function prototype

```
-(QAVResult)EnableAudioSend:(BOOL)enable;
```

Parameter	Type	Description
enable	BOOL	To enable audio upstreaming, set this parameter to `YES` ; otherwise, set it to `NO` .

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioSend:enabled];
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

Function prototype

```
-(BOOL)IsAudioSendEnabled;
```

Sample code

```
BOOL IsAudioSend = [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioSendEnabled];
```

Getting the real-time mic volume

This API is used to get the real-time mic volume. An int-type value in the range of 0-100 will be returned. It is recommended to call this API once every 20 ms.

This API is not applicable to the voice message service.

Function prototype

```
-(int)GetMicLevel;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetMicLevel];
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume. An int-type value in the range of 0-100 will be returned.

This API is not applicable to the voice message service.

Function prototype

```
-(int)GetSendStreamLevel();
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSendStreamLevel];
```

Setting the mic software volume

This API is used to set the mic volume. The corresponding parameter is `volume`, which is equivalent to attenuating or gaining the captured sound. 0 indicates that the audio is mute, while 100 indicates that the volume remains unchanged. The default value is 100.

This API is not applicable to the voice message service.

Function prototype

```
-(QAVResult)SetMicVolume:(int) volume;
```

Parameter	Type	Description
volume	int	Sets volume. Value range: 0-200

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] SetMicVolume:100];
```

Getting the mic software volume

This API is used to obtain the microphone volume. An "int" value is returned. Value 101 represents API SetMicVolume has not been called.

This API is not applicable to the voice message service.

Function prototype

```
-(int) GetMicVolume;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetMicVolume];
```

Voice Chat Playback APIs

Enabling or disabling the speaker

This API is used to enable/disable the speaker.

If accompaniment is used, please call this API as instructed in Accompaniment in Voice Chat.

EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv

Function prototype

```
-(void)EnableSpeaker:(BOOL)enable;
```

Parameter	Type	Description
isEnabled	boolean	To disable the speaker, set this parameter to `NO` , otherwise set it to `YES` .

Sample code

```
// Enable the speaker
[[[ITMGContext GetInstance] GetAudioCtrl] EnableSpeaker:YES];
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, while 1, on.

Function prototype

```
-(int)GetSpeakerState;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSpeakerState];
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

Function prototype

```
-(QAVResult)EnableAudioPlayDevice:(BOOL)enabled;
```

Parameter	Type	Description
enabled	BOOL	To disable a playback device, set this parameter to `NO` , otherwise set it to `YES` .

Sample code

```
// Enable the playback device
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioPlayDevice:enabled];
```

Getting the playback device status

This API is used to get the status of a playback device.

Function prototype

```
-(BOOL)IsAudioPlayDeviceEnabled;
```

Sample code

```
BOOL IsAudioPlayDevice =  [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioPlayDevi
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain mute. For more information on how to enable/disable the playback device, please see the `EnableAudioPlayDevice` API.

Function prototype

```
-(QAVResult)EnableAudioRecv:(BOOL)enabled;
```

Parameter	Type	Description
enabled	BOOL	To enable audio downstreaming, set this parameter to `YES` ; otherwise, set it to `NO` .

Sample code

```
[[[ITMGContext GetInstance]GetAudioCtrl ]EnableAudioRecv:enabled];
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

Function prototype

```
-(BOOL)IsAudioRecvEnabled;
```

Sample code

```
BOOL IsAudioRecv = [[[ITMGContext GetInstance] GetAudioCtrl] IsAudioRecvEnabled];
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume level. An int-type value will be returned to indicate the volume level. It is recommended to call this API once every 20 ms.

Function prototype

```
-(int)GetSpeakerLevel;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSpeakerLevel];
```

Getting the real-time downstreaming audio levels of other members in room

This API is used to get the real-time audio downstreaming volume of other members in the room. An int-type value will be returned. Value range: 0-200.

Function prototype

```
-(int)GetRecvStreamLevel:(NSString*) openID;
```

Parameter	Type	Description
openID	NSString	`openId` of another member in the room

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetRecvStreamLevel:(NSString*) openId
```

Dynamic setting of the volume of a member in the room

This API is used to set the volume of a member in the room and only applies to the local device.

Function prototype

```
-(int) SetSpeakerVolumeByOpenID:(NSString *)openId volume:(int)volume;
```

Parameter description

Parameter	Type	Description
openId	String *	OpenID that needs to adjust the volume
volume	int	Range: [0-200]; Default: 100

Getting the set volume percentage

Call this API to get volume set by SetSpeakerVolumeByOpenID

API prototype

```
-(int) GetSpeakerVolumeByOpenID:(NSString *)openId;
```

Returned values

API returns volume percentage set by OpenID, 100 by default.

Setting the speaker volume

This API is used to set the speaker volume.
The corresponding parameter is volume. 0 indicates that the audio is mute, while 100 indicates that the volume remains unchanged. The default value is 100.

Function prototype

```
-(QAVResult)SetSpeakerVolume:(int)vol;
```

Parameter	Type	Description
vol	int	Sets volume. Value range: 0-200

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] SetSpeakerVolume:100];
```

Getting the speaker volume

This API is used to get the speaker volume. An int-type value will be returned to indicate the volume. 101 indicates that the `SetSpeakerVolume` API has not been called.
"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

Function prototype

```
-(int)GetSpeakerVolume;
```

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] GetSpeakerVolume];
```


Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

Function prototype

```
-(QAVResult)EnableLoopBack:(BOOL)enable;
```

Parameter	Type	Description
enable	boolean	Specifies whether to enable in-ear monitoring.

Sample code

```
[[[ITMGContext GetInstance] GetAudioCtrl] EnableLoopBack:YES];
```

Modifying user's room audio type

This API is used to modify a user's room audio type. For the result, please see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

Function prototype

```
-(int)ChangeRoomType:(int)nRoomType;
```

Parameter	Type	Description
nRoomType	int	Target room type to be switched to. For room audio types, please see the `EnterRoom` API.

Sample code

```
[[[ITMGContext GetInstance]GetRoom ]ChangeRoomType:_roomType];
```

Getting user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, please see the `EnterRoom` API.

Function prototype

```
-(int)GetRoomType;
```

Sample code

```
[[[ITMGContext GetInstance]GetRoom ]GetRoomType];
```

Callback for modifying the room type

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result` , `error_info` , and `new_room_type` . The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	Indicates that the existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	Indicates that a user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	Indicates that a user is already in the room and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	Indicates that a room member calls the `ChangeRoomType` API to request a change of room audio type.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
```

```
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE:
          NSLog(@"ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE:%@ ",data);
          int result = ((NSNumber*)[data objectForKey:@"result"]).intValue;
          int newRoomType = ((NSNumber*) [data objectForKey:@"new_room_type"]).in
          int subEventType = ((NSNumber*) [data objectForKey:@"sub_event_type"]).
     }
}
```

Data details

Message	Data	Sa
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result;error_info;new_room_type;subEventType	{"e

The monitoring event of room call quality

The message for quality monitoring event triggered once every two seconds after room entry is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY` . The returned parameters include `weight` , `loss` , and `delay` , which represent the following information.

This API is used to monitor the network quality. If the user's network is poor, the business layer will remind the user to switch to a better network through the UI.

Parameter	Type	Description
weight	int	Value range: 1-50. 50 indicates excellent sound quality, 1 indicates very poor (barely usable) sound quality, and 0 represents an initial meaningless value. Generally, if the value is below 30, you can remind users that the network is poor and recommend them to switch the network.
loss	double	Upstream packet loss rate
delay	int	Voice chat delay in ms

Getting version number

This API is used to get the SDK version number for analysis.

Function prototype

```
-(NSString*)GetSDKVersion;
```

Sample code

```
[[ITMGContext GetInstance] GetSDKVersion];
```

Checking mic permission

This API is used to return the mic permission status.

Function prototype

```
-(ITMG_RECORD_PERMISSION)CheckMicPermission;
```

Parameter description

Parameter	Value	Description
ITMG_PERMISSION_GRANTED	0	Mic permission is granted.
ITMG_PERMISSION_Denied	1	Mic is disabled.
ITMG_PERMISSION_NotDetermined	2	No authorization box has been popped up to request the permission.
ITMG_PERMISSION_ERROR	3	An error occurred while calling the API.

Sample code

```
[[ITMGContext GetInstance] CheckMicPermission];
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

Function prototype

```
-(void)SetLogLevel:(ITMG_LOG_LEVEL)levelWrite (ITMG_LOG_LEVEL)levelPrint;
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
[[ITMGContext GetInstance] SetLogLevel:TMG_LOG_LEVEL_INFO TMG_LOG_LEVEL_INFO];
```

Setting the log printing path

This API is used to set the log printing path, which is /Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents by default.

Function prototype

```
-(void)SetLogPath:(NSString*)logDir;
```

Parameter	Type	Description
logDir	NSString	Path

Sample code

```
[[ITMGContext GetInstance] SetLogPath:Path];
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

Function prototype

```
-(NSString*)GetQualityTips;
```

Sample code

```
[[[ITMGContext GetInstance]GetRoom ] GetQualityTips];
```

Callback Messages

Message list

Message	Description
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	Indicates that a member enters an audio room.
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	Indicates that a member exits an audio room.
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	Indicates that a room is disconnected for network or other reasons.
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	Indicates a room type change event.
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Indicates that the room members are updated.
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	Indicates the room quality information.

Data list

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"",
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	result; error_info	{"error_info":"",
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"w

ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	result; error_info; sub_event_type; new_room_type	{"error_info":"",
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	result; error_info	{"deviceID":"{0 dd00542b47ae Audio)","error_
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	result; error_info	{"deviceID":"{0 dd00542b47ae Audio)","error_
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	result; error_info	{"deviceID":"{0 7e454093f229 Audio)","error_
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	result; error_info	{"deviceID":"{0 7e454093f229 Audio)","error_
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	user_list; event_id	{"event_id":1,"u
ITMG_MAIN_EVENT_TYPE_NUMBER_OF_USERS_UPDATE	AllUser; AccUser; ProxyUser	{"AllUser":3,"A
ITMG_MAIN_EVENT_TYPE_NUMBER_OF_AUDIOSTREAMS_UPDATE	AudioStreams	{"AudioStream
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	weight; loss; delay	{"weight":5,"los

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document describes how to integrating with and debug the GME APIs for macOS.

Note:

This document applies to GME SDK version 2.9.

Key Considerations for Using GME

GME provides two services: Voice chat service and voice messaging and speech-to-text service, both of which rely on key APIs such as Init and Poll.

Note:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.

Non-streaming speech-to-text API ***SpeechToText()***: There can be up to 10 concurrent requests per account.

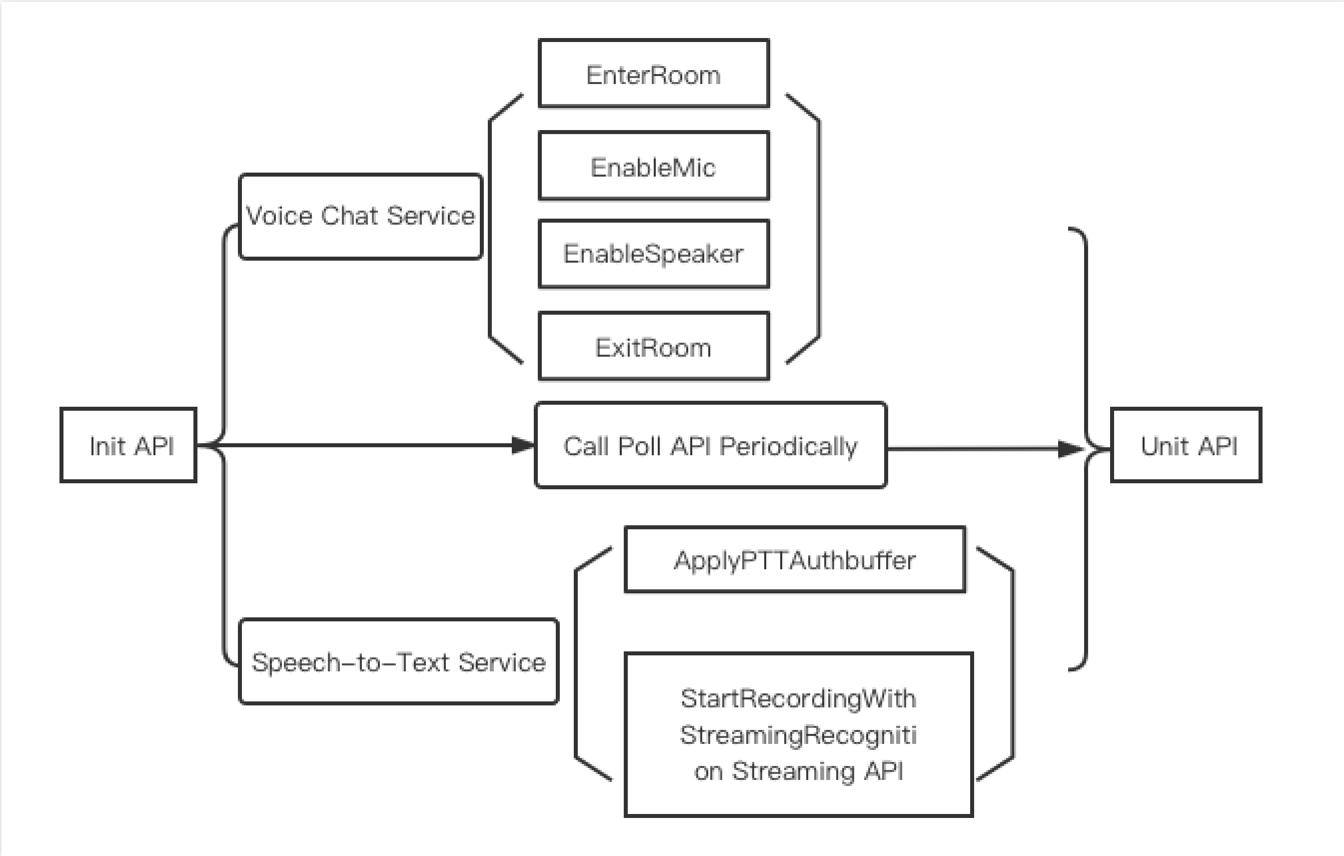
Streaming speech-to-text API ***StartRecordingWithStreamingRecognition()***: There can be up to 50 concurrent requests per account.

Real-time streaming speech-to-text API ***StartRealTimeASR()***: There can be up to 50 concurrent requests per account.

Note on Init API

If you need to use voice chat and voice messaging services at the same time, **you only need to call `Init` API once**.

The billing will not start after initialization. Receiving or sending a voice message in speech-to-text service is counted as a voice message DAU.



Directions

1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Initializing authentication, API: ApplyPTTAuthbuffer
4. Starting streaming speech recognition, API: StartRecordingWithStreamingRecognition
5. Stop recording, API: StopRecording
6. Uninitializing GME, API: UnInit

Important notes

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.

GME APIs should be called in the same thread.

The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error code, see Error Codes.

Core APIs

Before the initialization, the SDK is in the uninitialized status, and **you need to initialize it through the `Init` API** before you can use the voice chat and speech-to-text services.

Call the `Init` API before calling any APIs of GME.

If you have any questions when using the service, see General.

API	Description
InitEngine	Initializes GME
Poll	Triggers event callback
Pause	Pauses the system
Resume	Resumes the system
Uninit	Uninitializes GME
SetDefaultAudienceAudioCategory	Sets audio playback in background on device

Imported header files

```
#import "GMESDK/TMGEngine.h"
#import "GMESDK/QAVAuthBuffer.h"
```

Getting singleton

To use the voice feature, get the `ITMGContext` object first.

```
+ (ITMGContext*) GetInstance;
```

Sample code

```
//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
```

Setting callbacks

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages.

Sample code

`ITMGDelegate` is used for declaration.

```
@interface TMGDemoViewController ()<ITMGDelegate>{}
```

```
ITMGDelegate < NSObject >

//TMGSampleViewController.m
ITMGContext* _context = [ITMGContext GetInstance];
_context.TMGDelegate = [DispatchCenter getInstance];
```

The API callback messages is processed in `OnEvent` . For the message type, see `ITMG_MAIN_EVENT_TYPE` . The message content is a dictionary for parsing the API callback contents.

Function prototype

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data;
```

Sample code

```
//TMGRealTimeViewController.m
TMGRealTimeViewController ()< ITMGDelegate >


- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data {
    NSString *log = [NSString stringWithFormat:@"OnEvent:%d,data:%@", (int)eventTyp
    [self showLog:log];
    NSLog(@"====%@====", log);
    switch (eventType) {
        // Step 6/11 : Perform the enter room event
        case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM: {
            int result = ((NSNumber *)[data objectForKey:@"result"]).intValue;
            NSString *error_info = [data objectForKey:@"error_info"];

            [self showLog:[NSString stringWithFormat:@"OnEnterRoomComplete:%d msg:(

            if (result == 0) {
                [self updateStatusEnterRoom:YES];
            }
        }
        break;
    }
}

// Refer to DispatchCenter.h and DispatchCenter.m
```

Initializing SDK

This API is used to initialize the GME service. It is recommended to call it when initializing the application. No fee is incurred for calling this API.

For more information on how to get the `sdkAppID` parameter, see Activating Services.
The openID uniquely identifies a user with the rules stipulated by the application developer and unique in the application (currently, only INT64 is supported).

Note:

The Init API must be called in the same thread with other APIs. It is recommended to call all APIs in the main thread.

Function prototype

```
-(int)InitEngine:(NSString*)sdkAppID openID:(NSString*)openID;
```

Parameter	Type	Description
sdkAppId	String	`AppId` provided by the GME service from the Tencent Cloud console
OpenId	String	`OpenId` can only be in Int64 type, which is passed after being converted to a string.

Returned Value	Description
QAV_OK= 0	Initialized SDK successfully.
QAV_ERR_SDK_NOT_FULL_UPDATE= 7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is only a reminder but will not cause an initialization failure.
If this error is reported during integration, please check the integrity and version of the SDK file as prompted.
If this error is returned after executable file export, please ignore it and try to avoid displaying it in the UI.

Sample code

```
_openId = _userIdText.text;
_appId = _appIdText.text;
[[ITMGContext GetInstance] InitEngine:SDKAPPID openID:_openId];
```

Triggering event callback

Event callbacks can be triggered by periodically calling the `Poll` API in `update`. The `Poll` API should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run exceptionally.
For more information, see the EnginePollHelper.m file in Demo.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

Function prototype

```
-(void)Poll;
```

Sample code

```
[[ITMGContext GetInstance] Poll];
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause.
If you need to pause the audio when switching to the background, you can call the `Pause` API in the listening code used to switch to the background, and call the `Resume` API in the listening event used to resume the foreground.

Function prototype

```
-(QAVResult)Pause;
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

Function prototype

```
-(QAVResult)Resume;
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **Switching accounts requires uninitialization**.

Function prototype

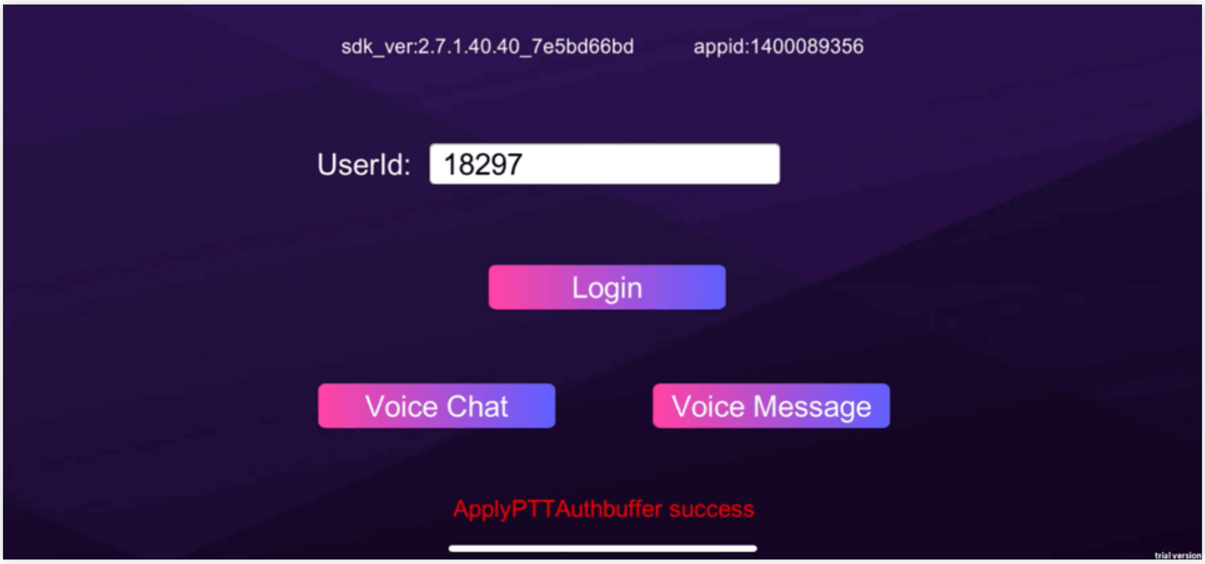
```
-(int)Uninit;
```

Sample code

```
[[ITMGContext GetInstance] Uninit];
```


Voice Messaging and Speech-to-Text

Voice messaging refers to recording and sending a voice message. At the same time, the voice message can be converted to text and translated, as shown below:

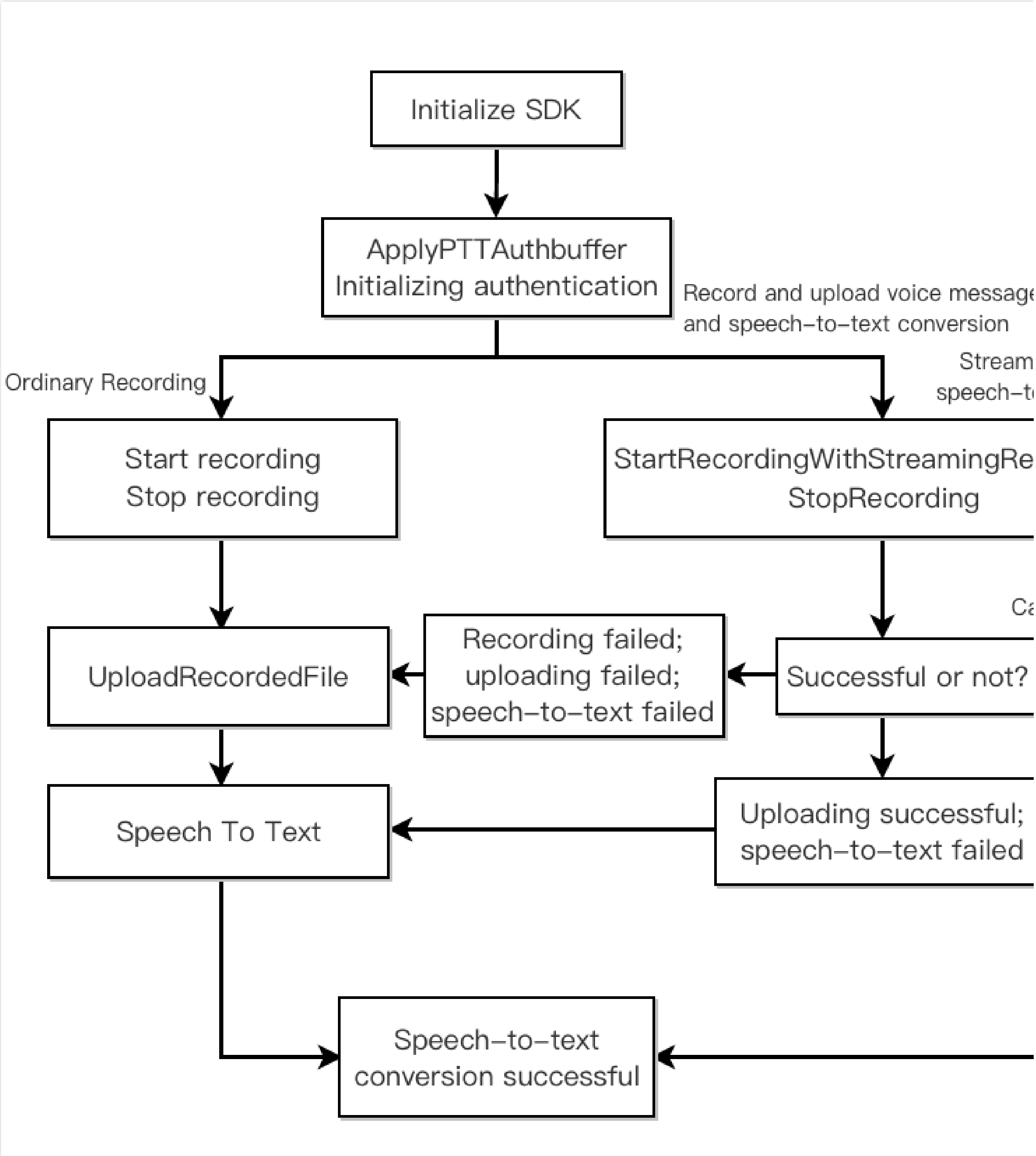


Note:

It is recommended to use the streaming speech-to-text service.

You do not need to enter a voice chat room when using the voice message service.

Voice messaging and speech-to-text conversion flowchart



Integrating Voice Messaging and Speech-to-Text Service

Voice messaging and speech-to-text APIs

API	Description

ApplyPTTAuthbuffer	Initializes authentication
SetMaxMessageLength	Specifies the maximum length of voice message
StartRecording	Starts recording
StartRecordingWithStreamingRecognition	Starts streaming recording
PauseRecording	Pauses recording
ResumeRecording	Resumes recording
StopRecording	Stops recording
CancelRecording	Cancels recording
GetMicLevel	Gets the real-time mic volume
SetMicVolume	Sets the recording volume
GetMicVolume	Gets the recording volume
GetSpeakerLevel	Gets the real-time speaker volume
SetSpeakerVolume	Sets the playback volume
GetSpeakerVolume	Gets the playback volume
UploadRecordedFile	Uploads the audio file
DownloadRecordedFile	Downloads the audio file
PlayRecordedFile	Plays back audio
StopPlayFile	Stops playing back audio
GetFileSize	Gets the audio file size
GetVoiceFileDuration	Gets the audio file duration
SpeechToText	Converts speech to text

Maximum recording duration

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, please call the `SetMaxMessageLength` API to set it after initialization.

Initializing the SDK

Before the initialization, the SDK is in the uninitialized status, and you need to initialize it through the `Init` API before you can use the voice chat and voice message services.
If you have any questions when using the service, see Speech-to-text Conversion.

Authentication information

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, please use the backend deployment key as detailed in Authentication Key.
To get authentication for voice message and speech-to-text, the room ID parameter must be set to `null` .

Function prototype

```
@interface QAVAuthBuffer : NSObject
+ (NSData*) GenAuthBuffer:(unsigned int)appId roomId:(NSString*)roomId openID:(NSSt
+ @end
```

Parameter	Type	Description
appId	int	`AppId` from the Tencent Cloud console.
roomId	NSString	Enter `null` .
openID	NSString	User ID, which is the same as `openID` during initialization.
key	NSString	Permission key from the Tencent Cloud console.

Sample code

```
#import "GMESDK/QAVAuthBuffer.h"
NSData* authBuffer = [QAVAuthBuffer GenAuthBuffer:SDKAPPID3RD.intValue roomId:_room
```

Initializing authentication

Call authentication initialization after initializing the SDK. For more information on how to get the `authBuffer` , see `genAuthBuffer` (the voice chat authentication information API).

Function prototype

```
public abstract int ApplyPTTAuthbuffer(byte[] authBuffer);
```

Parameter	Type	Description

authBuffer	NSData*	Authentication

Sample code

```
[[[ITMGContext GetInstance]GetPTT]ApplyPTTAuthbuffer:(NSData *)authBuffer];
```

Streaming Speech Recognition

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call `StopRecording`**. The callback will be returned after the recording is stopped.

Function prototype

```
-(int)StartRecordingWithStreamingRecognition:(NSString *)filePath;
-(int)StartRecordingWithStreamingRecognition:(NSString *)filePath language:(NSStrin
```

Parameter	Type	Description
filePath	String	Path of stored audio file
speechLanguage	String	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List
translateLanguage	String	The language into which the audio file will be translated. For parameters, see Language Parameter Reference List (This parameter is currently unavailable. Enter the same value as that of `speechLanguage`)

Sample code

```
recordfilePath = [docDir stringByAppendingFormat:@"/test_%d.ptt",index++];
[[[ITMGContext GetInstance] GetPTT] StartRecordingWithStreamingRecognition:recordfi
```

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen for callback messages in the callback function `onEvent` . Event messages are divided into:
`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.
`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real-time during the recording, which is equivalent to returning the recognized text while speaking.
The event message will be identified in the `OnEvent function` based on the actual needs. The passed parameters include the following four messages.

Message Name	Description
result	A return code for judging whether the streaming speech recognition is successful.
text	Text converted from speech
file_path	Local path of stored recording file
file_id	Backend URL address of recording file, which will be retained for 90 days. `fileid` is fixed at `http://gme-v2-`

Note:
The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Suggested Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.

Sample code

```
- (void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary*)data
{
    NSNumber *number = [data objectForKey:@"result"];
    switch (eventType)
    {
        case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                donwLoadUrlPath = data[@"file_id"];

                recordfilePath = [data objectForKey:@"file_path"];
                _localFileField.text = recordfilePath;

                _donwloadUrlField.text = [data objectForKey:@"file_id"] ;

                UITextField *_audiotoTextField =(UITextField*)objc_getAssociatedObj
                _audiotoTextField.text = [data objectForKey:@"text"] ;
            }

        }
            break;
    }
}
```

Voice Message Recording

The recording process is as follows: start recording -> stop recording -> return recording callback -> start the next recording.

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

Function prototype

```
-(QAVResult)SetMaxMessageLength:(int)msTime
```

Parameter	Type	Description
msTime	int	Audio duration in ms. Value range: 1000 < msTime <= 58000

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SetMaxMessageLength:(int)msTime];
```

Starting recording

This API is used to start recording. The recording file must be uploaded first before you can perform operations such as speech-to-text conversion. **To stop recording, call** `StopRecording` .

Function prototype

```
-(int)StartRecording:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of stored audio file

Sample code

```
recordfilePath =[docDir stringByAppendingFormat:@"/test_%d.ptt",index++];
[[[ITMGContext GetInstance]GetPTT]StartRecording:recordfilePath];
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

Function prototype

```
-(QAVResult)StopRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]StopRecording];
```

Callback for recording start

A callback will be executed through a delegate function to pass a message when recording is completed.

To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.

The callback function `OnEvent` will be called after recording is started. The event message `ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameter includes `result` and `file_path` .

Error codes

Error Code Value	Cause	Suggested Solution
4097	Parameter is empty.	Check whether the API parameters in the code are correct.
4098	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
4099	Recording is in progress.	Ensure that the SDK recording feature is used at the right time.
4100	Audio data is not captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	Mic permission is required for using the SDK. To add the permission, see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
        {
        //Recording callback
```

```
        }
            break;
    }
}
```

Pausing recording

This API is used to pause recording. If you want to resume recording, please call the `ResumeRecording` API.

Function prototype

```
-(int)PauseRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]PauseRecording;
```

Resuming recording

This API is used to resume recording.

Function prototype

```
-(int)ResumeRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]ResumeRecording;
```

Canceling recording

This API is used to cancel recording. There is no callback after cancellation.

Function prototype

```
-(QAVResult)CancelRecording;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]CancelRecording];
```

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT`.

Function prototype

```
-(QAVResult)GetMicLevel;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetMicLevel];
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT`.

Function prototype

```
-(QAVResult)SetMicVolume:(int) volume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SetMicVolume:100];
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT`.

Function prototype

```
-(int)GetMicVolume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetMicVolume];
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT`.

Function prototype

```
-(QAVResult)GetSpeakerLevel;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetSpeakerLevel];
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice messaging. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT`.

Function prototype

```
-(QAVResult)SetSpeakerVolume:(int)volume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SetSpeakerVolume:100];
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice messaging. An int-type value will be returned. Value range: 0-200.

Note:

This API is different from the voice chat API and is in `ITMGPTT` .

Function prototype

```
-(int)GetSpeakerVolume;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetSpeakerVolume];
```

Voice Message Playback

Playing back audio

This API is used to play back audio.

Function prototype

```
-(int)PlayRecordedFile:(NSString*)filePath;
-(int)PlayRecordedFile:(NSString*)filePath VoiceType:(ITMG_VOICE_TYPE) type;
```

Parameter	Type	Description
downloadFilePath	NSString	Local audio file path
type	ITMG_VOICE_TYPE	Voice changer type. For more information, see Voice Changing Effects.

Error codes

Error Code Value	Cause	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]PlayRecordedFile:path];
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameter includes `result` and `file_path` .

Error codes

Error Code Value	Cause	Suggested Solution
20481	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Parameter is empty.	Check whether the API parameters in the code are correct.
20484	Internal error.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE:
        {
        // Callback for audio playback
        }
            break;
    }
}
```


Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

Function prototype

```
-(int)StopPlayFile;
```

Sample code

```
[[[ITMGContext GetInstance]GetPTT]StopPlayFile];
```

Getting audio file size

This API is used to get the size of an audio file.

Function prototype

```
-(int)GetFileSize:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of audio file, which is a local path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetFileSize:path];
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

Function prototype

```
-(int)GetVoiceFileDuration:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of audio file, which is a local path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]GetVoiceFileDuration:path];
```

Voice Message Upload and Download

Uploading an audio file

This API is used to upload an audio file.

Function prototype

```
-(void)UploadRecordedFile:(NSString*)filePath;
```

Parameter	Type	Description
filePath	NSString	Path of uploaded audio file, which is a local path.

Sample code

```
[[[ITMGContext GetInstance]GetPTT]UploadRecordedFile:path];
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameters include `result` , `file_path` , and `file_id` .

Error codes

Error Code Value	Cause	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.

8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
8195	A network error occurred.	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                _donwloadUrlField.text = [data objectForKey:@"file_id"] ;
                donwLoadUrlPath = [data objectForKey:@"file_id"] ;
            }
        }
            break;
    }
}
```

Downloading the audio file

This API is used to download an audio file.

Function prototype

```
-(void)DownloadRecordedFile:(NSString*)fileId downloadFilePath:(NSString*)downloadF
```


Parameter	Type	Description
fileID	NSString	File URL path
downloadFilePath	NSString	Local path of saved file

Sample code

```
[[[ITMGContext GetInstance]GetPTT]DownloadRecordedFile:fileIdpath downloadFilePath:
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result`, `file_path`, and `file_id`.

Error codes

Error Code Value	Cause	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
12291	Network storage system exception	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	Server file system error.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed during the process of getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.

12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.
12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                _audiofileToPlayField.text = [data objectForKey:@"file_path"] ;
                donwLoadLocalPath = [data objectForKey:@"file_path"];
            }
            else
            {
                donwLoadLocalPath = NULL;
            }
        }
            break;
    }
}
```

Speech-to-Text Service

Converting audio file to text

This API is used to convert a specified audio file to text.

Function prototype

```
-(void)SpeechToText:(NSString*)fileID;
```

Parameter	Type	Description
fileID	NSString	URL of audio file

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SpeechToText:fileID];
```

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Function prototype

```
-(void)SpeechToText:(NSString*)fileID (NSString*)speechLanguage (NSString*)translat
```

Parameter	Type	Description
fileID	NSString*	URL of audio file, which will be retained on the server for 90 days
speechLanguage	NSString*	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translateLanguage	NSString*	The language into which the audio file will be translated. For parameters, see Language Parameter Reference List. This parameter is currently unavailable. Enter the same value as that of `speechLanguage` .

Sample code

```
[[[ITMGContext GetInstance]GetPTT]SpeechToText:fileID speechLanguage:"cmn-Hans-CN"
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` and `text` (recognized text).

Error codes

Error Code Value	Cause	Suggested Solution

32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.
32784	Incorrect speech-to-text conversion parameter.	Check whether the API parameter `fileid` in the code is empty.
32785	Speech-to-text translation returned an error.	Error with the backend of voice message and speech-to-text feature. Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.

Sample code

```
-(void)OnEvent:(ITMG_MAIN_EVENT_TYPE)eventType data:(NSDictionary *)data{
    NSLog(@"OnEvent:%lu,data:%@",(unsigned long)eventType,data);
    switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
        {
            if (data != NULL &&[[data objectForKey:@"result"] intValue]== 0)
            {
                UITextField *_audiotoTextField =(UITextField*)objc_getAssociatedObj
                _audiotoTextField.text = [data objectForKey:@"text"] ;
            }
        }
            break;
    }
}
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for analysis.

Function prototype

```
-(NSString*)GetSDKVersion;
```

Sample code

```
[[ITMGContext GetInstance] GetSDKVersion];
```

Checking mic permission

This API is used to return the mic permission status.

Function prototype

```
-(ITMG_RECORD_PERMISSION)CheckMicPermission;
```

Parameter description

Parameter	Value	Description
ITMG_PERMISSION_GRANTED	0	Mic permission is granted.
ITMG_PERMISSION_Denied	1	Mic is disabled.
ITMG_PERMISSION_NotDetermined	2	No authorization box has been popped up to request the permission.
ITMG_PERMISSION_ERROR	3	An error occurred while calling the API.

Sample code

```
[[ITMGContext GetInstance] CheckMicPermission];
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

Function prototype

```
-(void)SetLogLevel:(ITMG_LOG_LEVEL)levelWrite (ITMG_LOG_LEVEL)levelPrint;
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written. `TMG_LOG_LEVEL_NONE` indicates not to write. Default value: TMG_LOG_LEVEL_INFO
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed. `TMG_LOG_LEVEL_NONE` indicates not to print. Default value: TMG_LOG_LEVEL_ERROR

ITMG_LOG_LEVEL

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
[[ITMGContext GetInstance] SetLogLevel:TMG_LOG_LEVEL_INFO TMG_LOG_LEVEL_INFO];
```

Setting the log printing path

. This API is used to set the log printing path, which is /Users/username/Library/Containers/xxx.xxx.xxx/Data/Documents by default.. Call before Init.

Function prototype


```
-(void)SetLogPath:(NSString*)logDir;
```

Parameter	Type	Description
logDir	NSString	Path

Sample code

```
[[ITMGContext GetInstance] SetLogPath:Path];
```

Callback Messages

Message list

Message	Description
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Indicates that PTT recording is completed.
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Indicates that PTT upload is completed.
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Indicates that PTT download is completed.
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Indicates that PTT playback is completed.
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Indicates that speech-to-text conversion is completed.

Data list

Message	Data	Sample
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	result; file_path	{"file_path":
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	result; file_path;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	result; file_path;file_id	{"file_id":"",'

ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	result; file_path	{"file_path":
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	result; text;file_id	{"file_id":"",'
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	result; file_path; text;file_id	{"file_id":"",'

H5 SDK

Project Configuration

최종 업데이트 날짜: : 2024-01-18 15:13:51

This document describes how to get started with the GME APIs for HTML5.

Features Supported by the SDK for HTML5

Feature	Supported by HTML5
Basic voice chat features	Yes
Sound effect and accompaniment	Partially supported
3D voice	No
Range voice	No
Room management	No
Audio forwarding routing	No
Custom message channel	No
Voice messaging	No
Speech-to-text	No

Platforms Supported by the SDK for HTML5

OS/Platform	Browser/WebView	Version Requirement	Remarks
iOS	Safari	11.1.2	Safari on a later version
	Chrome	-	iOS currently doesn't support Chrome.
	Weixin browser	-	iOS currently doesn't support the Weixin browser.

Android	TBS (the default WebView of Weixin and Mobile QQ)	43600	The default built-in browser kernel of Weixin and Mobile QQ is TBS
	Chrome	60+	H.264 support is required
macOS	Chrome	47+	-
	Safari	11+	-
Windows (PC)	Chrome	52+	-
	QQ Browser	10.2	-

Preparations

The SDK can be obtained in the following steps:

1. Go to the SDK Download Guide page.
2. Locate the SDK resources for HTML5 on the page.
3. Click **Download**.

Frontend Project Configuration Steps

1. Open ports

If you have configured a firewall across your network, be sure to open the following ports:

Protocol	Port
TCP	8687
UDP	8000, 8800, 443

Import the SDK by using CDN.

2. Import frontend library files

Import `WebRTCService.min.js` into the project as demonstrated in the following code:

```
<head>
    <script src="../dist/WebRTCService.min.js"></script>
</head>
```

3. Add the audio tag

```
<div id="gme-audio-wrap"></div>
```

Server-Side Deployment Steps

The use of the GME SDK requires authentication which involves keys and is not suitable for implementation on the client. You are recommended to deploy it separately.
If only client implementation is needed for the time being, please refer to the provided demo project.

1. Download the program

Download the sample `authBuffer` program, which can sign the authentication information for a specified `SDKAppID` .

2. Configure the server-side authentication project

Go to the `signdemo` directory and modify the `config.js` file: open the `config.js` file, delete the default configuration, and call the `appidMap` function in the place where the code is deleted (the parameters are the `SDKAppid` applied for on the Tencent Cloud backend and the corresponding authentication key).

```
const AuthBufferConfig = function () {
    this.appidMap = {};
    this.appidMap["1400089356"] = "1cfbfd2a1a03a53e";
};
// Replace `1400089356` with the `sdkAppid` applied for on the Tencent Cloud backen
```

Note:

The `AuthKey` must correspond to your `SDKAppid` .

3. Deploy the server-side authentication project

Go to the directory where the sample `authBuffer` program resides and run the following statement to install the dependencies:

```
npm i
```

Then, execute the `node index.js` script to run the signature service.

Note:

As the async syntax is used, make sure that your node is v8 or later. Run `node -v` on the command line to view the version.

4. Test the deployment result

You can run the following command on the command line for test (make sure that your system has a `curl` command):

```
// Generate a `userSig`:
curl "http://127.0.0.1:10005/" --data "sdkappid=1400089356&roomid=1234123&openid=12
```

After the signing program is executed, the authentication information will be returned as shown below:

```
{"userSig":"AqhHE7QHLFYPfV/zfyrdRYHfuUn6eOA8g/J6GMjVy//Shr5ByJPTi8hzR2KyXMvn","erro
```

API Documentation

최종 업데이트 날짜: : 2024-01-18 15:13:51

Note:
Only some GME features are supported by the SDK for HTML5. Please refer to this document for the supported APIs and evaluate whether the SDK for HTML5 is appropriate for your business scenario.

API	Description
Init	Initializes API
SetTMGDelegate	Sets delegation
EnterRoom	Enters audio room
EnableMic	Turns on/off the capturing device
EnableSpeaker	Turns on/off the playback device
SetMicVolume	Sets mic volume
ExitRoom	Exits audio room

Note:
After a GME API is called successfully, `QAVError.OK` will be returned with the value being 0.
Authentication is required for room entry in GME. For more information, see the authentication section in relevant documentation.
Operation on devices should be performed after successful room entry.
Starting from Chrome 74, `navigator.mediaDevices` can only be used in an HTTPS environment; therefore, please use HTTPS.

Integrating JQ

You need to integrate JQ to use the demo.

```
<!--Step 2: Add the audio container-->
<!--Container, which is used to carry audio tags and cannot be omitted.-->
<div id="gme-audio-wrap"></div>
```

Initialization APIs

Before initialization, the SDK is in the uninitialized state. A room can be entered only after the initialization authentication is performed and the SDK is initialized.

Initializing the SDK

For more information on how to get parameters, see Access Guide.

This API requires the `SDKAppID` from the Tencent Cloud console and the `openId` as parameters. The `openId` uniquely identifies a user with the rules stipulated by the application developer and must be unique in the application (currently, only INT64 is supported).

Note:

The SDK must be initialized before a user can enter a room.

Function prototype

```
WebGMEAPI.fn.Init = function (document, SdkAppId, openId) {...}
```

Parameter	Description
document	HTML DOM Document object
SdkAppId	`SdkAppId` from the Tencent Cloud console
openId	Developer-defined user account with a value greater than 10,000, which is used to identify the user.

Sample code

```
const cSdkAppId = () => document.getElementById("input-SdkAppId").value;
const cOpenID = () => document.getElementById("input-OpenID").value;
gmeAPI.Init(document, cSdkAppId(), cOpenID());
```

Setting callbacks

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK to receive callback messages. The callback function should be registered to the SDK before room entry.

Function prototype

```
WebGMEAPI.fn.SetTMGDelegate = function (delegate){...}
```

Parameter	Description
onEvent	SDK callback event

Sample code

```
gmeAPI.SetTMGDelegate(onEvent);
```

Voice Chat APIs

You should initialize and call the SDK to enter a room before voice chat can start.

Entering a room

When a user enters a room with the generated authentication information, the `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` message will be received as a callback. Mic and speaker are not turned on by default after room entry.

Function prototype

```
WebGMEAPI.fn.EnterRoom = function (roomId, roomType, authBuffer) {...}
```

Parameter	Description
roomId	Room ID, which can contain up to 127 characters
roomType	Room audio type
authBuffer	Authentication key. For more information on how to get it, see Project Configuration.

Sample code

```
function bindButtonEvents() {
       $("#start_btn").click(function () {
           console.log('start!');
           // Step 1: Get the `AuthBuffer`
           var FetchSigCgi = 'http://134.175.146.244:10005/';
           $.ajax({
               type: "POST",
               url: FetchSigCgi,
               dataType: 'json',
               data: {
```

```
                sdkappid: cSdkAppId(),
                roomid: cRoomNum(),
                openid: cOpenID(),
            },
            success: function (json) {
                // Step 2: `AuthBuffer` is obtained successfully
                if (json && json.errorCode === 0) {
                    let userSig = json.userSig;
                    gmeAPI.Init(document, cSdkAppId(), cOpenID());
                    gmeAPI.SetTMGDelegate(onEvent);
                    gmeAPI.EnterRoom(cRoomNum(), 1, userSig);
                } else {
                    console.error(json);
                }
            },
            error: function (err) {
                console.error(err);
            }
        });
    });
```

Event Callbacks

After the user enters the room, the message `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` will be sent and identified in the `OnEvent` function.

Sample code

```
onEvent = function (eventType, result) {
        if (eventType === gmeAPI.event.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
       {
           // Entered room successfully
       }
       else if (eventType === gmeAPI.event.ITMG_MAIN_EVENT_TYPE_USER_UPDATE)
       {
           app._data.downStreamInfoList = result.PeerInfo;// Received peer informa
           app._data.brSend = result.UploadBRSend;// Bitrate of the uploaded audio
           app._data.rtt = result.UploadRTT;// Upload RTT
       }
       else if (eventType === gmeAPI.event.ITMG_MAIN_EVENT_TYPE_EXIT_ROOM)
       {
           // Exited room successfully
       }
       else if (eventType === gmeAPI.event.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT)
       {
           // Room disconnected
```

```
        }
    };
```

The received peer information is as follows (downStreamInfoList):

Parameter	Description
brRecv	The received bitrate
delay	Receipt delay
jitterBufferMs	Delay caused by jitter
jitterReceived	The received Jitter

Exiting a room

This API is called to exit the current room. It is an async API. There will be a callback after room exit. The returned value of `AV_OK` indicates a successful async delivery.

Function prototype

```
WebGMEAPI.fn.ExitRoom = function (){...}
```

Sample code

```
gmeAPI.ExitRoom();
```

Turning on/off the mic

This API is used to turn on/off the mic. Mic and speaker are not turned on by default after room entry.

Function prototype

```
WebGMEAPI.fn.EnableMic = function (bEnable) {...}
```

Parameter	Description
isEnabled	To turn on the mic, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
gmeAPI.EnableMic(false);
```

Setting the mic volume

This API is used to set the mic volume. The corresponding parameter is `volume` . 0 indicates that the audio is mute, while 100 indicates that the volume remains unchanged. The default value is 100.

Function prototype

```
WebGMEAPI.fn.SetMicVolume = function (volume){...}
```

Parameter	Description
volume	Sets the volume. Value range: 0-100.

Sample code

```
gmeAPI.SetMicVolume(100);
```

Turning on/off the speaker

This API is used to turn on/off the speaker.

Function prototype

```
WebGMEAPI.fn.EnableSpeaker = function (bEnable){...}
```

Parameter	Description
isEnabled	To turn off the speaker, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
gmeAPI.EnableSpeaker(true);
```

Electron SDK

Integrating SDK

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document describes how to configure an Electron project for the GME APIs for Electron.

Supported Platforms

Windows

Importing the SDK

Step 1: Install Node.js

1. Download the latest version of the Node.js installation package `Windows Installer (.msi) 64-bit` for Windows.
2. Open `Node.js command prompt` in the application list and open a terminal window.

Step 2. Install Electron

Run the following command in the terminal to install Electron. V4.0.0 or later is recommended.

```
$ npm install electron -g
```

Step 3. Install the GME SDK for Electron

1. Use the following npm command in your Electron project to install the GME SDK:

```
$ npm install gme-electron-sdk@latest --save
```

2. In the project script, import and use the module:

```
const { GmeContext } = require('gme-electron-sdk');
// import gmesdk from 'gme-electron-sdk';
gmeContext = new GmeContext();
// Get the SDK version number
gmeContext.GetSDKVersion();
```

Step 4. Create an executable program

Install the packaging tool. We recommend you use Electron Forge. You can run the following command:

1. Add Electron Forge to your application's development dependencies and run the `import` command to set the scaffold of Forge:

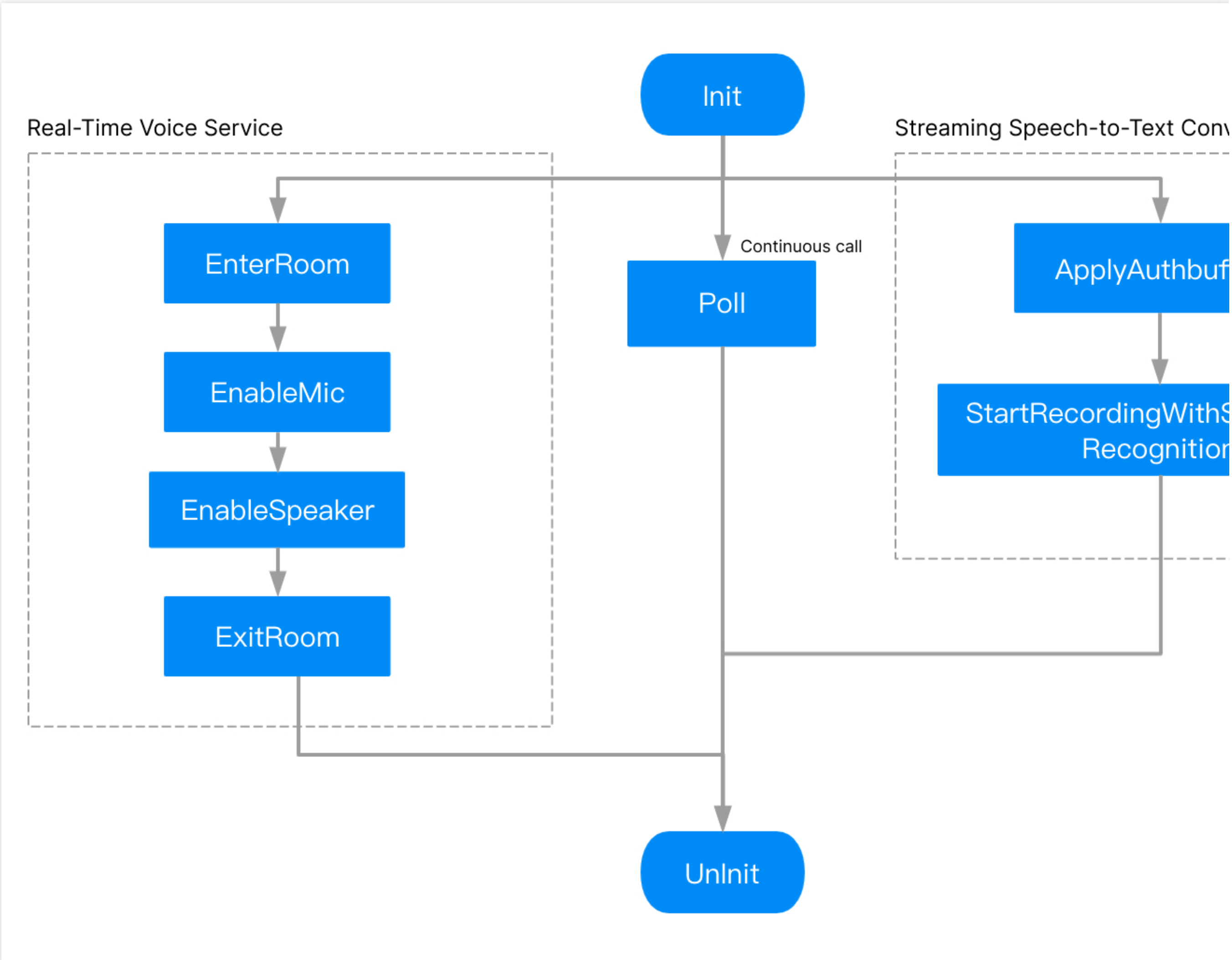
```
npm install --save-dev @electron-forge/cli
npx electron-forge import
```

2. Run the `make` command of Forge to create a distributable application:

```
npm run make
```

Voice Chat

최종 업데이트 날짜: : 2024-01-18 15:15:48



This document describes how to integrate with and debug GME client APIs for the voice chat feature for Electron.

Key Considerations for Using GME

GME provides the voice chat, voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll` .

Notes

You have created a GME application and obtained the SDK `AppID` and key. For more information, see Activating Services.
You have activated **GME voice chat, voice messaging, and speech-to-text services**. For more information, see Activating Services.
Configure your project before using GME; otherwise, the SDK will not take effect.
After a GME API is called successfully, `GmeError.AV_OK` will be returned with the value being `0` .
GME APIs should be called in the same thread.
The `Poll` API should be called periodically for GME to trigger event callbacks.
For detailed error codes, see Error Codes.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:
1. Initializing GME
2. Calling `Poll` periodically to trigger callbacks
3. Entering a voice chat room
4. Turning on the mic
5. Turning on the speaker
6. Exiting the voice chat room
7. Uninitializing GME

Core APIs

API	Description
Init	Initializes GME.
Poll	Triggers an event callback.
Pause	Pauses the system.
Resume	Resumes the system.
Uninit	Uninitializes GME.

Importing the GME module


```
const { GmeContext } = require('gme-electron-sdk');
```

Getting an instance

To use the voice chat feature, get the `GmeSDK` object first.

```
context = new GmeContext();
```

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice chat, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
//class GmeSDK
Init(appid: string, openid: string): number;
```

Parameter	Type	Description
sdkAppId	string	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	string	`openID` can only be in Int64 type, which is passed in after being converted to a string. You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
GmeError.AV_OK= 0	The SDK was initialized successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Check whether the SDK file is complete. We recommend that you delete it and then import it again.

Note on 7015 error code

The error code 7015 is identified based on the MD5 value. If this error is reported during integration, check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party enhancement, the Unity packaging mechanism, and other factors, the MD5 value of the library file will be affected, resulting in misjudgment. Therefore, **ignore this error in the logic for official releases**, and avoid displaying it on the UI.

Sample code

```
string SDKAPPID3RD = "14000xxxxx";
string openId="10001";
number ret = context.Init(SDKAPPID3RD, openId);
// Determine whether the initialization is successful by the returned value
if (ret != GmeError.AV_OK)
{
        console.log("Failed to initialize the SDK:");
        return;
}
```

Setting callbacks

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages before room entry.

Function prototype and sample code

Register the callback function to the SDK for receiving callback messages before room entry.

```
SetTMGDelegate(cb: ITMGDelegate);
// When initializing the SDK
context =  GmeSDK.GetInstance();
context.setTMGDelegate(function(eventId, msg){
  if (type == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
  {
            // Processing callbacks
  }
});
```

Triggering event callback

You need to periodically call the `Poll` API to trigger event callbacks. The `Poll` API is GME's message pump and should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run abnormally. For more information, see the `EnginePollHelper` file in SDK Download Guide.

Call the `Poll` API periodically

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype


```
Poll():number;
```

Sample code

```
setInterval(function () {
      context.Poll();
    }, 50);
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. For example, when the application switches to the background (OnApplicationPause, isPause=True), and you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
Pause() :number
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
Resume() :number
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game account is bound to** `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.

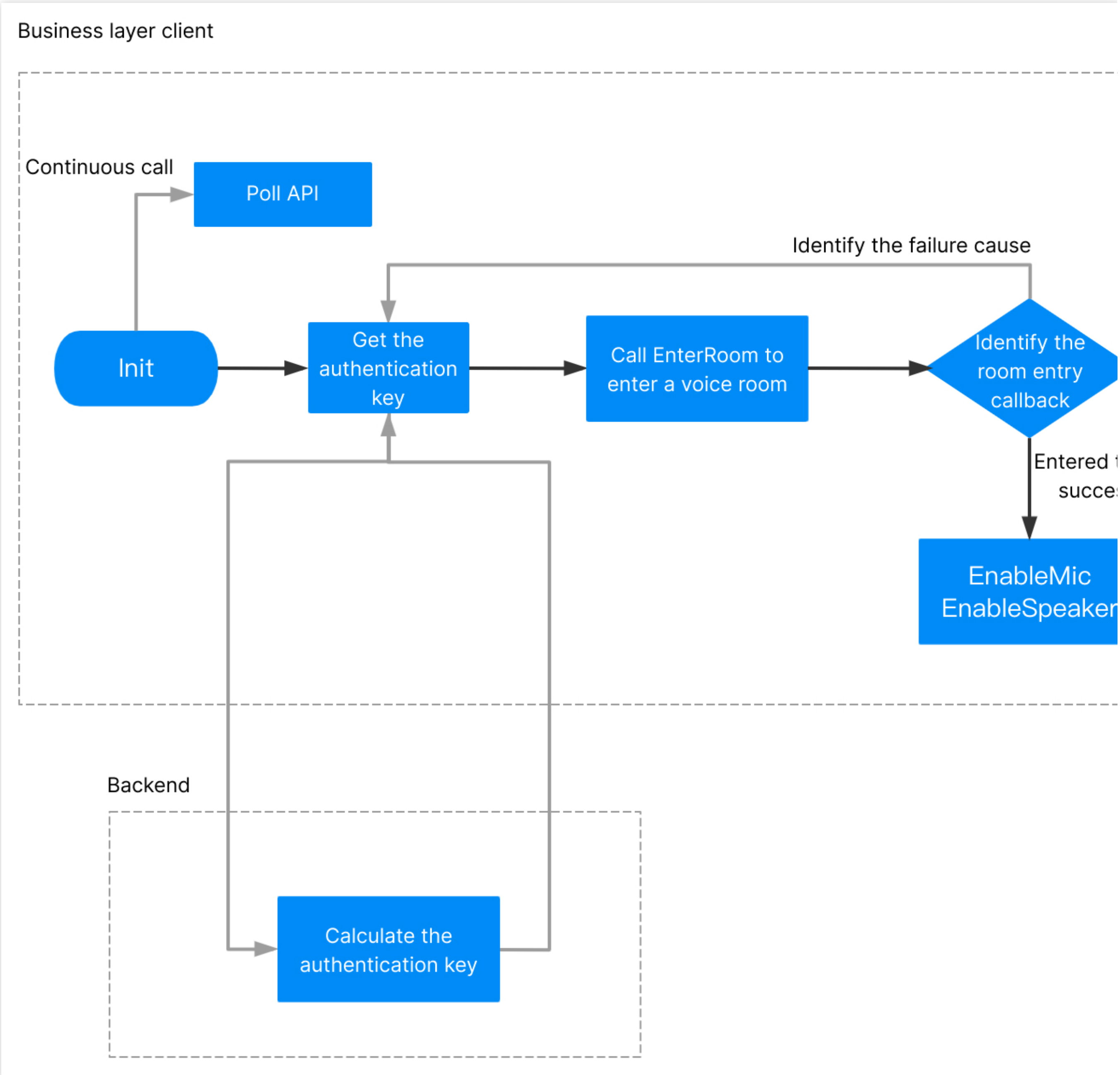
API prototype

```
Uninit() : number;
```

Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, see Sound and Audio.



API	Description
GenAuthBuffer	Calculates the local authentication key.
EnterRoom	Enters a room.
ExitRoom	Exits a room.
IsRoomEntered	Determines whether room entry is successful.

Local authentication key calculation

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.

API prototype

```
GenAuthBuffer(appId: string,roomId: string, openId:string, appKey: number) :string;
```

Parameter	Type	Description
appId	string	`AppID` from the Tencent Cloud console
roomId	string	Room ID, which can contain up to 127 characters.
openId	string	User ID, which is the same as `openID` during initialization.
key	number	Permission key from the Tencent Cloud console.

Sample code

```
let userSig = context.GenAuthBuffer(this.appid, this.roomId, this.userId, this.aut
context.EnterRoom(this.roomId, this.roomType, userSig);
```

Entering a room

This API is used to enter a room with the generated authentication information. The mic and speaker are not enabled by default after room entry.

Note:

If the room entry callback result is `0` , the room entry is successful. If `0` is returned from the `EnterRoom` API, it doesn't necessarily mean that the room entry is successful.

The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will become smooth sound quality. Only when a member in the room calls the `ChangeRoomType` API, the audio type of the room will be changed.

API prototype

```
EnterRoom(roomid: string, roomType: number, appKey: string) :number;
```

Parameter	Type	Description

roomId	string	Room ID, which can contain up to 127 characters.
roomType	ITMGRoomType	Room type. We recommend that you select `ITMG_ROOM_TYPE_FLUENCY` for games. For more information on room audio types, see Sound Quality.
appKey	string	Authentication key

Sample code

```
context.EnterRoom(roomID, ITMG_ROOM_TYPE_STANDARD, retAuthBuff);
```

Callback for room entry

After the user enters the room, the `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` event type will be called back to notify the room entry result, which can be listened on for processing. A successful callback means that the room entry is successful, and the billing **starts**.

Billing references:

Purchase Guide

Billing

Will the billing continue if the client is disconnected from the server when using the voice chat?

Sample code

```
// Listen on an event:
 gmeContext.setTMGDelegate(function(eventId, msg){
  switch (eventId) {
      case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
      {
      }
  }
});
```

Data details

Message	Data	Example
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnection callback notification `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` . At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` . When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` .

Error codes

Error Code	Cause and Suggested Solution
7006	Authentication failure causes: `AppID` doesn't exist or is incorrect. An error occurred while authenticating `authbuff` . Authentication expired. `OpenId` is invalid.
7007	The user was already in another room.
1001	The user was already in the process of entering a room but repeated this operation. We recommend that you not call the room entry API until the room entry callback is returned.
1003	The user was already in the room and called the room entry API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally.

Exiting a room

This API is used to exit the current room. It is an async API. The returned value `AV_OK` indicates a successful async delivery. If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API; instead, you can directly call the `EnterRoom` API.

API prototype

```
ExitRoom(): number;
```

Sample code

```
context.ExitRoom();
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM` . The sample code is shown below:

Sample code

```
gmeContext.setTMGDelegate(function(eventId, msg){
  switch (eventId) {
      case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
      {
             // Process
        break;
      }
  }
});
```

Determining whether user has entered room

This API is used to determine whether the user has entered a room. A value in boolean type will be returned. Do not call this API during room entry.

API prototype

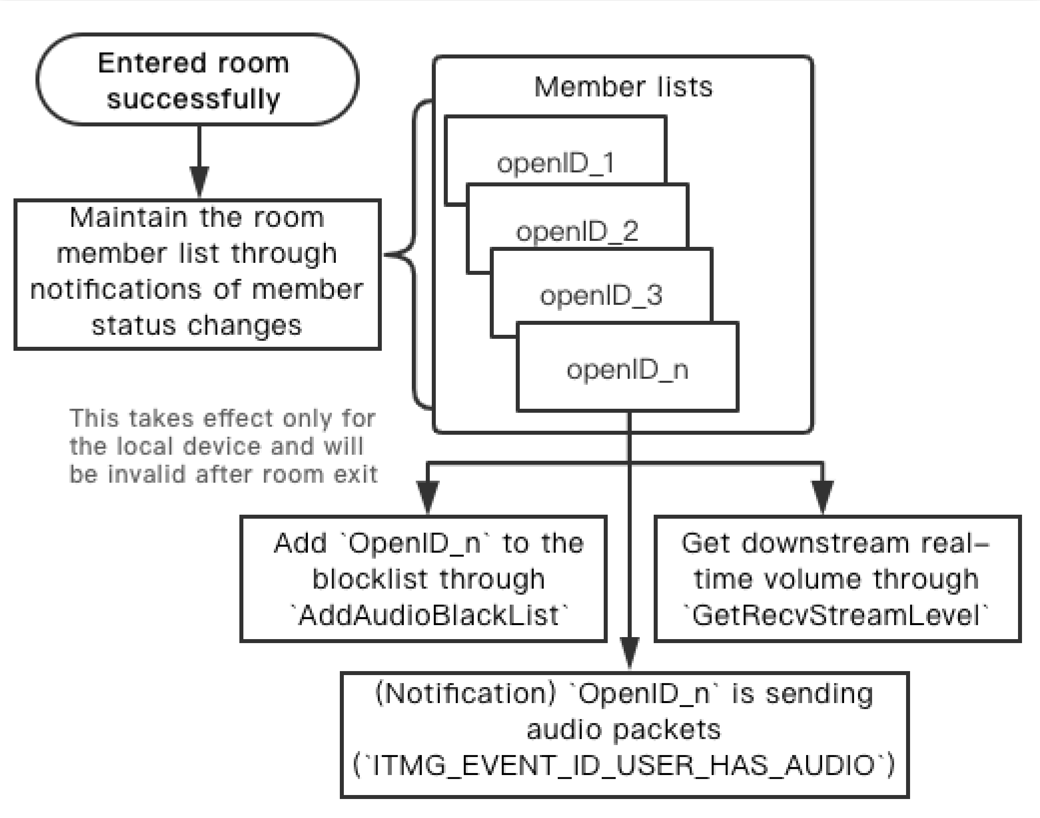
```
IsRoomEntered() :boolean
```

Sample code

```
context.IsRoomEntered();
```

Room Status Maintenance

APIs in this section are used to display speaking members and members entering or exiting the room and mute a member in the room at the business layer.



API/Notification	Description
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The member status changed.
AddAudioBlackList	Mutes a member in the room.
RemoveAudioBlackList	Unmutes a member.
IsOpenIdInAudioBlackList	Queries whether the user of the specified `openid` is muted.

Notification events of the member room entry and speaking status

This event is used to get the user speaking in the room and display the user on the UI, and to send a notification when someone enters or exits the room.

A notification for this event will be sent only when the status changes. To get the member status in real time, cache the notification when it is received at the business layer. The event message `ITMG_MAIN_EVNET_TYPE_USER_UPDATE` containing `event_id`, `count`, and `openIdList` will be returned, which will be identified in the `OnEvent` notification.

Notifications of the `EVENT_ID_ENDPOINT_NO_AUDIO` audio event will be sent only when the threshold is exceeded; that is, other members in the room can receive the notification that the local user stops speaking only after the local client captures no voice for two seconds.

The audio event returns only the member speaking status but not the specific volume level. If you need the specific volume levels of members in the room, you can use the `GetVolumeById` API.

event_id	Description	Maintenance
EVENT_ID_ENDPOINT_ENTER	Return the `openid` of the member entering the room.	Member list
EVENT_ID_ENDPOINT_EXIT	Return the `openid` of the member exiting the room.	Member list
EVENT_ID_ENDPOINT_HAS_AUDIO	Return the `openid` of the member sending audio packets in the room. This event can be used to determine whether a user is speaking and display the voiceprint effect.	Chat member list
EVENT_ID_ENDPOINT_NO_AUDIO	Return the `openid` of the member stopping sending audio packets in the room.	Chat member list

Sample code

```
context.setTMGDelegate(function(eventId, msg){
  if (type == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
  {
               // Process
            switch (eventID)
             {
             case EVENT_ID_ENDPOINT_ENTER:
                  // A member enters the room
                  break;
             case EVENT_ID_ENDPOINT_EXIT:
                  // A member exits the room
                break;
            case EVENT_ID_ENDPOINT_HAS_AUDIO:
                // A member sends audio packets
                break;
            case EVENT_ID_ENDPOINT_NO_AUDIO:
                // A member stops sending audio packets
                break;

            default:
                break;
             }
        break;
  }
});
```


Muting a member in the room

This API is used to add an ID to the audio data blocklist. This operation blocks audio from someone and only applies to the local device without affecting other devices. The returned value `0` indicates that the call is successful.

Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B.

If B blocks neither A nor C, B can hear both of them.

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

API prototype

```
AddAudioBlackList(openId: string) :number
```

Parameter	Type	Description
openId	string	`openid` of the user to be blocked

Sample code

```
context.AddAudioBlackList(openId);
```

Unmuting

This API is used to remove an ID from the audio data blacklist. A returned value of 0 indicates the call is successful.

API prototype

```
RemoveAudioBlackList(openId: string) :number
```

Parameter	Type	Description
openId	string	ID to be unblocked

Sample code

```
context.RemoveAudioBlackList(openId);
```

Querying whether a user is muted

This API is used to query whether an ID is blocked. The returned value `true` indicates that the ID is blocked, while `false` indicates not.

API prototype

```
IsOpenIdInAudioBlackList(openId: string) :boolean
```

Parameter	Type	Description
openId	string	ID to be queried

Sample code

```
boolean isInBlackList = context.IsOpenIdInAudioBlackList(openId);
```

Voice Chat Capturing APIs

The voice chat APIs can only be called after SDK initialization and room entry.
When the user clicks the button of enabling/disabling the mic or speaker on the UI, we recommend that you call the `EnableMic` or `EnableSpeaker` API.
To enable the user to press the mic button on the UI to speak and release it to stop speaking, we recommend that you call `EnableAudioCaptureDevice` once during room entry and call `EnableAudioSend` to enable the user to speak while pressing the button.

API	Description
EnableMic	Enables/Disables the mic.
GetMicState	Gets the mic status.
EnableAudioCaptureDevice	Enables/Disables the capturing device.
IsAudioCaptureDeviceEnabled	Gets the capturing device status.
EnableAudioSend	Enables/Disables audio upstreaming.
IsAudioSendEnabled	Gets the audio upstreaming status.
GetMicLevel	Gets the real-time mic volume level.
GetSendStreamLevel	Gets the real-time audio upstreaming volume level.
SetMicVolume	Sets the mic volume level.

GetMicVolume	Gets the mic volume level.

Enabling or disabling mic

This API is used to enable/disable the mic. The mic and speaker are not enabled by default after room entry.

EnableMic = EnableAudioCaptureDevice + EnableAudioSend

API prototype

```
EnableMic(bEnable: boolean) : number
```

Parameter	Type	Description
isEnabled	boolean	To enable the mic, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
// Turn on mic
context.EnableMic(true);
```

Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

API prototype

```
GetMicState() :number
```

Sample code

```
context.GetMicState();
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.

This API can only be called after room entry. The device will be disabled automatically after room exit.

Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

API prototype

```
EnableAudioCaptureDevice(enable:boolean) :number
```


Parameter	Type	Description
enable	boolean	To enable the capturing device, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable capturing device
context.EnableAudioCaptureDevice(true);
```

Getting the capturing device status

This API is used to get the status of a capturing device.

API prototype

```
IsAudioCaptureDeviceEnabled():boolean
```

Sample code

```
boolean IsAudioCaptureDevice = context.IsAudioCaptureDeviceEnabled();
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain muted. For more information on how to enable/disable the capturing device, see the `EnableAudioCaptureDevice` API.

API prototype

```
EnableAudioSend(bEnable: boolean) :number
```

Parameter	Type	Description
isEnabled	boolean	To enable audio upstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
context.EnableAudioSend(true);
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

API prototype

```
IsAudioSendEnabled():boolean
```

Sample code

```
boolean IsAudioSend = context.IsAudioSendEnabled();
```

Getting the real-time mic volume

This API is used to get the real-time mic volume level. A number-type value in the range of 0–100 will be returned. We recommend that you call this API once every 20 ms.

API prototype

```
GetMicLevel():number
```

Sample code

```
context.GetMicLevel();
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume level. A number-type value in the range of 0–100 will be returned.

API prototype

```
GetSendStreamLevel() :number
```

Sample code

```
context.GetSendStreamLevel();
```

Setting the mic software volume

This API is used to set the mic volume level. The corresponding parameter is `volume` , which is equivalent to attenuating or gaining the captured sound.

API prototype

```
SetMicVolume(volume:number) :number
```

Parameter	Type	Description
volume	number	Value range: 0-200. Default value: `100` . `0` indicates that the audio is muted, while `100` indicates that the volume level remains unchanged.

Sample code

```
number micVol = (value * 100);
context.SetMicVolume (micVol);
```

Getting the mic software volume

This API is used to get the mic volume level. A number-type value will be returned. `101` indicates that the `SetMicVolume` API has not been called.

API prototype

```
GetMicVolume()
```

Sample code

```
context.GetMicVolume();
```

Voice Chat Playback APIs

API	Description
EnableSpeaker	Enables/Disables the speaker.
GetSpeakerState	Gets the speaker status.
EnableAudioPlayDevice	Enables/Disables the playback device.
IsAudioPlayDeviceEnabled	Gets the playback device status.
EnableAudioRecv	Enables/Disables audio downstreaming.
IsAudioRecvEnabled	Gets the audio downstreaming status.

GetSpeakerLevel	Gets the real-time speaker volume level.
GetRecvStreamLevel	Gets the real-time downstreaming audio levels of other members in the room.
SetSpeakerVolume	Sets the speaker volume level.
GetSpeakerVolume	Gets the speaker volume level.

Enabling or disabling speaker

This API is used to enable/disable the speaker. **EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv**

API prototype

```
EnableSpeaker(bEnable: boolean) : number;
```

Parameter	Type	Description
bEnable	boolean	To disable the speaker, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
// Turn on the speaker
context.EnableSpeaker(true);
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, and 1 is on.

API prototype

```
GetSpeakerState() :number
```

Sample code

```
context.GetSpeakerState();
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

API prototype

```
EnableAudioPlayDevice(enable:boolean) :number
```

Parameter	Type	Description
enable	boolean	To disable the playback device, set this parameter to `false` ; otherwise, set it to `true` .

Sample code

```
context.EnableAudioPlayDevice(true);
```

Getting the playback device status

This API is used to get the status of a playback device.

API prototype

```
IsAudioPlayDeviceEnabled() :boolean
```

Sample code

```
boolean enable = context.IsAudioPlayDeviceEnabled();
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain muted. For more information on how to enable/disable the playback device, see the `EnableAudioPlayDevice` API.

API prototype

```
EnableAudioRecv(bEnable: boolean) :number
```

Parameter	Type	Description
isEnabled	boolean	To enable audio downstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
context.EnableAudioRecv(true);
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

API prototype

```
IsAudioRecvEnabled():boolean
```

Sample code

```
boolean IsAudioRecv = context.IsAudioRecvEnabled();
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume level. A number-type value will be returned to indicate the volume level. We recommend that you call this API once every 20 ms.

API prototype

```
GetSpeakerLevel():number
```

Sample code

```
context.GetSpeakerLevel();
```

Getting the real-time downstreaming audio levels of other members in room

This API is used to get the real-time audio downstreaming volume levels of other members in the room. A number-type value will be returned. Value range: 0–200.

API prototype

```
GetRecvStreamLevel(openId: string) :number
```

Parameter	Type	Description
openId	string	`openId` of another member in the room

Sample code

```
number level =GetRecvStreamLevel(openId);
```

Dynamically setting the volume of a member of the room

This API is used to set the volume of a member in the room. It takes effect only on the local.

API prototype

```
SetSpeakerVolumeByOpenID(openId: string, volume:number) :number;
```

Parameter	Type	Description
openId	string	`OpenID` of the target user
volume	number	Percentage. Recommended value range: 0–200. Default value: `100` .

Sample code

```
context.SetSpeakerVolumeByOpenID(openId,  100);
```

Getting volume percentage

This API is used to get the volume level set by `SetSpeakerVolumeByOpenID` .

API prototype

```
GetSpeakerVolumeByOpenID(openId: string) :number;
```

Parameter	Type	Description
openId	string	`OpenID` of the target user

Returned values

API returns volume percentage set by OpenID, where 100 is by default.

Sample code

```
context.GetSpeakerVolumeByOpenID(openId);
```

Setting the speaker volume

This API is used to set the speaker volume.

API prototype

```
SetSpeakerVolume(volume:number) :number
```

Parameter	Type	Description
volume	number	Volume level. Value range: 0–200. Default value: `100`. `0` indicates that the audio is muted, while `100` indicates that the volume level remains unchanged.

Sample code

```
number vol = 100;
context.SetSpeakerVolume(vol);
```

Getting the speaker volume

This API is used to get the speaker volume. A number-type value will be returned to indicate the volume. `101` indicates that the `SetSpeakerVolume` API has not been called.
"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

API prototype

```
GetSpeakerVolume() :number
```

Sample code

```
numbet volume = context.GetSpeakerVolume();
```

Device Selection APIs

Device selection APIs can be used only on PC.

API	Description
GetMicListCount	Gets the number of mics.
GetMicList	Enumerates mics.
GetSpeakerListCount	Gets the number of speakers.
GetSpeakerList	Enumerates speakers.

SelectMic	Selects a mic.
SelectSpeaker	Selects a speaker.

Getting the number of mics

This API is used to get the number of mics.

Function prototype

```
GetMicListCount() :number
```

Sample code

```
var micListCount = context.GetMicListCount();
```

Enumerating mics

This API is used together with the `GetMicListCount` API to enumerate mics.

Function prototype

```
GetMicList() :GmeAudioDeviceInfo[];
```

Sample code

```
var micList = context.GetMicList();
```

Selecting mic

This API is used to select a mic. If this API is not called or `DEVICEID_DEFAULT` is passed in, the default mic will be selected.
The 0th device id returned in the GetMicList API is the default device of the call device. If there is a selected call device, it will be maintained by service. If it is unplugged, the call device will be changed back into the default device.

Function prototype

```
SelectMic(micId: string) :number;
```

Parameter	Type	Description
micId	string	Mic ID, which is from the list returned by `GetMicList` .

Sample code

```
context.SelectMic(deviceID);
```

This API is used to get the number of speakers.

Function prototype

```
GetSpeakerListCount() :number;
```

Sample code

```
context.GetSpeakerListCount();
```

Enumerating speakers

This API is used together with the `GetSpeakerListCount` API to enumerate speakers.

Function prototype

```
GetSpeakerList(): GmeAudioDeviceInfo[]
```

Sample code

```
var speakList = GetSpeakerList();
```

Selecting speaker

This API is used to select a playback device. If this API is not called or `DEVICEID_DEFAULT` is passed in, the default playback device will be selected.

Function prototype

```
SelectSpeaker(speakerId: string) :number
```

Parameter	Type	Description
speakerId	string	Speaker ID, which is from the list returned by `GetSpeakerList` .

Sample code

```
var ret = SelectSpeaker(deviceID);
```

Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

API prototype

```
EnableLoopBack(bEnable: boolean) :number
```

Parameter	Type	Description
enable	boolean	Specifies whether to enable.

Sample code

```
context.EnableLoopBack(true);
```

Getting user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, please see the `EnterRoom` API.

API prototype

```
GetRoomType() :number
```

Sample code

```
context.GetRoomType();
```

Changing the room type

This API is used to modify a user's room audio type. For the result, please see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

API prototype

```
ChangeRoomType(roomType: number) :number
```

Parameter	Type	Description
roomtype	number	Room type to be switched to. For room audio types, see the `EnterRoom` API.

Sample code

```
context.ChangeRoomType(ITMG_ROOM_TYPE_FLUENCY);
```

Callback event

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result`, `error_info`, and `new_room_type`. The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	The existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	A user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	A user is already in the room, and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	A room member calls the `ChangeRoomType` API to request a change of the room audio type.

Sample code

```
context.setTMGDelegate(function(eventId, msg){
  if (ITMGContext.ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE == typ
        {
        // Process room type events
      }
});
```

The monitoring event of room call quality

This is the quality monitoring event used to listen on the network quality. If your network conditions are poor, the business layer will ask you to switch the network through the UI. This event is triggered once every two seconds after room entry, and its message is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY` . The returned parameters include `weight` , `loss` , and `delay` , which are as detailed below:

Parameter	Type	Description
weight	number	Value range: 1–50. `50` indicates excellent sound quality, `1` indicates very poor (barely usable) sound quality, and `0` represents an initial meaningless value. Generally, if the value is below 30, you can remind users that the network is poor and recommend them to switch the network.
Loss	var	Upstream packet loss rate
Delay	number	Voice chat delay in ms

Getting version number

This API is used to get the SDK version number for analysis.

API prototype

```
GetSDKVersion() :string
```

Sample code

```
context.GetSDKVersion();
```

Setting the application name and version

This API is used to set the application name and version.

API prototype

```
SetAppVersion(appVersion: string) : number
```

Parameter description

Parameter	Type	Description
appVersion	string	Application name and version

Sample code

```
context.SetAppVersion("gme V2.0.0");
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
SetLogLevel(level: number) : number
```

Parameter description

Parameter	Type	Description
level	number	Sets the log level. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO` .

`level` description:

level	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
context.SetLogLevel(TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

Platform	Path
Windows	%appdata%\\GMEGLOBAL\\GME\\ProcessName

API prototype

```
SetLogPath(logPath: string)
```

Parameter	Type	Description
logPath	string	Path

Sample code

```
string logDir = ""// Set a path by yourself
context.SetLogPath(logDir);
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

API prototype

```
GetQualityTips() :string
```

Sample code

```
string tips = context.GetQualityTips();
```

Callback message

Message	Description	Data
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room.	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room.	result; error
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected due to a network or another issue.	result; error

ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated.	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	The reconnection to the room started.	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	The reconnection to the room succeeded.	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched.	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status was changed.	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started.	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped.	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker was changed.	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A new speaker was added.	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker was lost.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A new mic was added.	result; error

ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic was lost.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic was changed.	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room network quality changed.	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Voice message recording was completed.	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Voice message upload was completed.	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Voice message download was completed.	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Voice message playback was completed.	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast speech-to-text conversion was completed.	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion	result; file_p text;file_id

	was completed.	
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	Streaming speech-to-text conversion is in progress.	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed.	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed.	result; text;file_id

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document describes how to integrate with and debug GME client APIs for the voice messaging and speech-to-text services for Electron.

Key Considerations for Using GME

GME provides the real-time voice service and voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

Notes

You have created a GME application and obtained the SDK `AppID` and key. For more information, see Activating Services.
You have activated **GME real-time voice service and voice messaging and speech-to-text services**. For more information, see Activating Services.
Configure your project before using GME; otherwise, the SDK will not take effect.
After a GME API is called successfully, `GmeError.AV_OK` will be returned with the value being `0`.
GME APIs should be called in the same thread.
The `Poll` API should be called periodically for GME to trigger event callbacks.
For detailed error codes, see Error Codes.

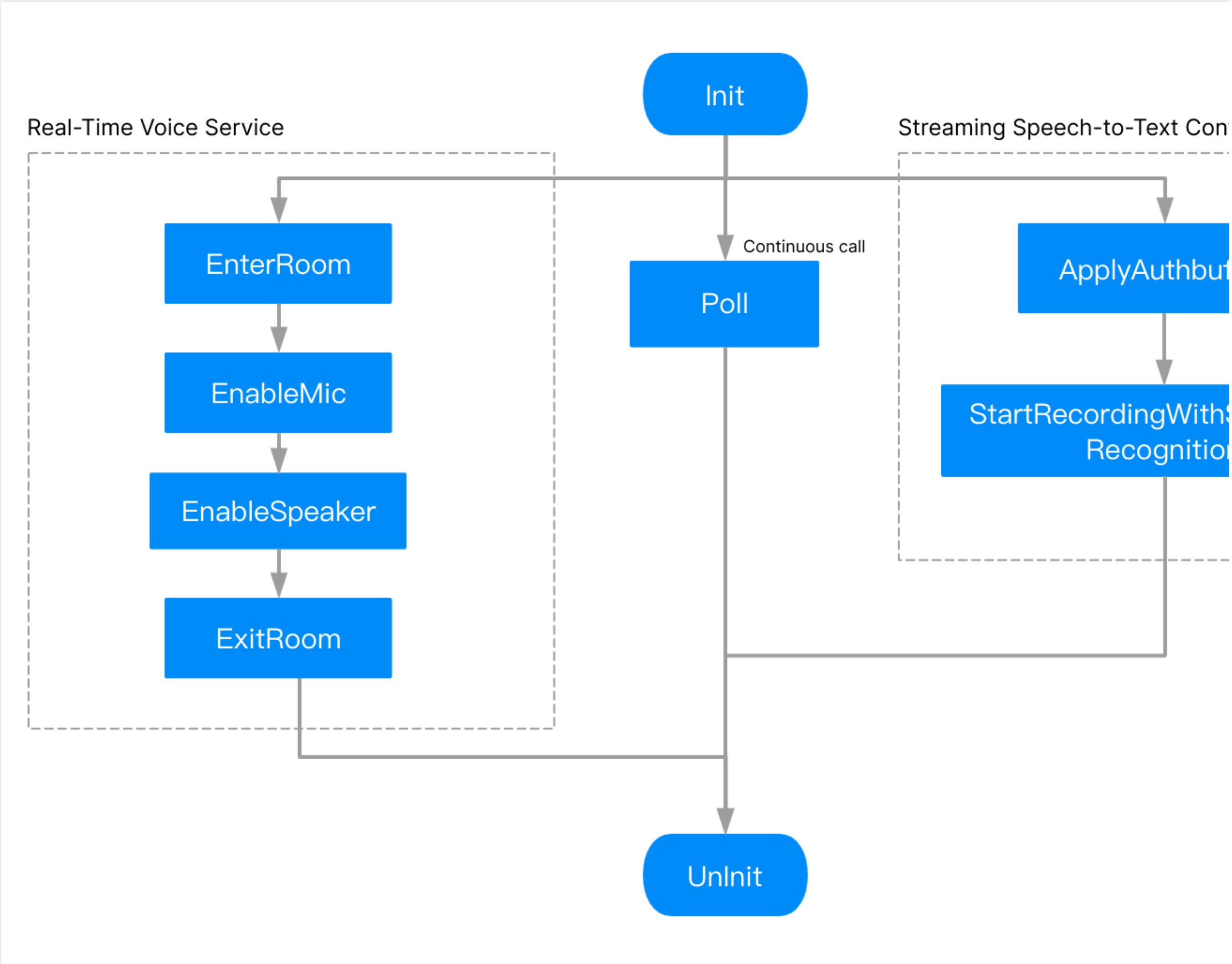
Note:

There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.
Non-streaming speech-to-text API **PttSpeechToText()**: There can be up to 10 concurrent requests per account.
Streaming speech-to-text API **PttStartRecordingWithStreamingRecognition()**: There can be up to 50 concurrent requests per account.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME, API: Init
2. Calling Poll periodically to trigger event callbacks, API: Poll
3. Initializing authentication, API: ApplyPTTAuthbuffer
4. Starting streaming speech recognition, API: PttStartRecordingWithStreamingRecognition
5. Stopping recording, API: PttStopRecording
6. Uninitializing GME, API: UnInit

TS class

```
`GmeContext`: GME business implementation APIs
`GmeError`: GME error code definition class
```

Core APIs

API	Description
Init	Initializes GME
Poll	Triggers event callback
Uninit	Uninitializes GME

Importing the GME module

```
const { GmeContext } = require('gme-electron-sdk');
```

Getting an instance

var m_context = new GmeContext();

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
Init(appid: string, openid: string): number;
```

Parameter	Type	Description
sdkAppId	string	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	string	`openID` can only be in `Int64` type, which is passed in after being converted to a string. You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
GmeError.AV_OK= 0	Initialized the SDK successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Checks whether the SDK file is complete. It is recommended to delete it and then import the SDK again.

Notes on 7015 error code

The 7015 error code is judged by MD5. If this error is reported during integration, please check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Sample code

```
number ret = m_context.Init(sdkAppId, openID);
// Determine whether the initialization is successful by the returned value
if (ret != GmeError.AV_OK)
{
     console.log("Failed to initialize the SDK:");
        return;
}
```

Triggering event callback

Event callbacks can be triggered by calling the `Poll` API in the timer. The `Poll` API is GME's message pump and should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run abnormally. For more information, see the `EnginePollHelper` file in SDK Download Guide.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
Poll();
```

Sample code

```
setInterval(function () {
      m_context.Poll();
    }, 50);
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

API prototype

```
Uninit() : number
```

Voice Messaging and Speech-to-Text Services

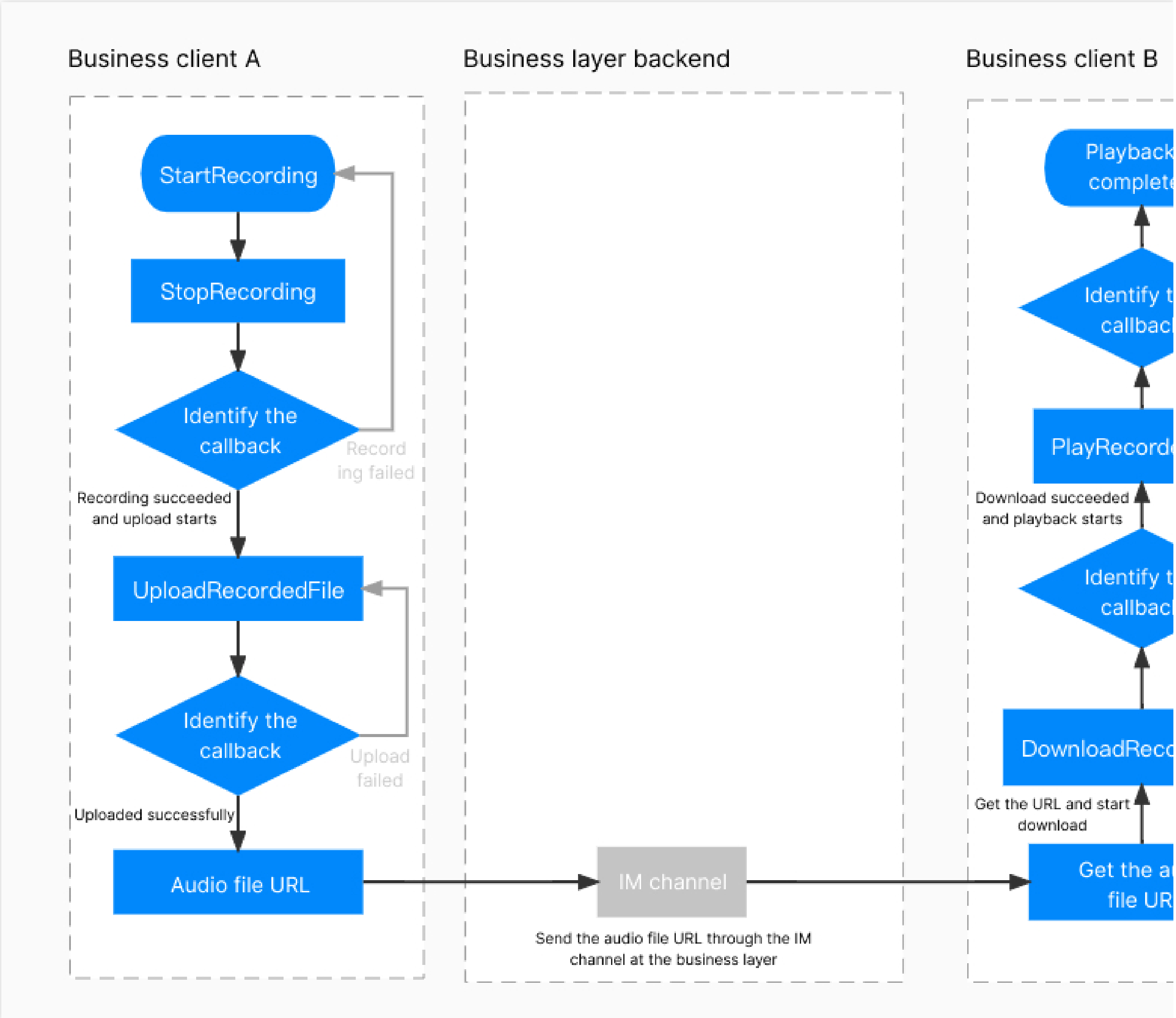
Note:

The speech-to-text service consists of fast recording-to-text conversion and streaming speech-to-text conversion.

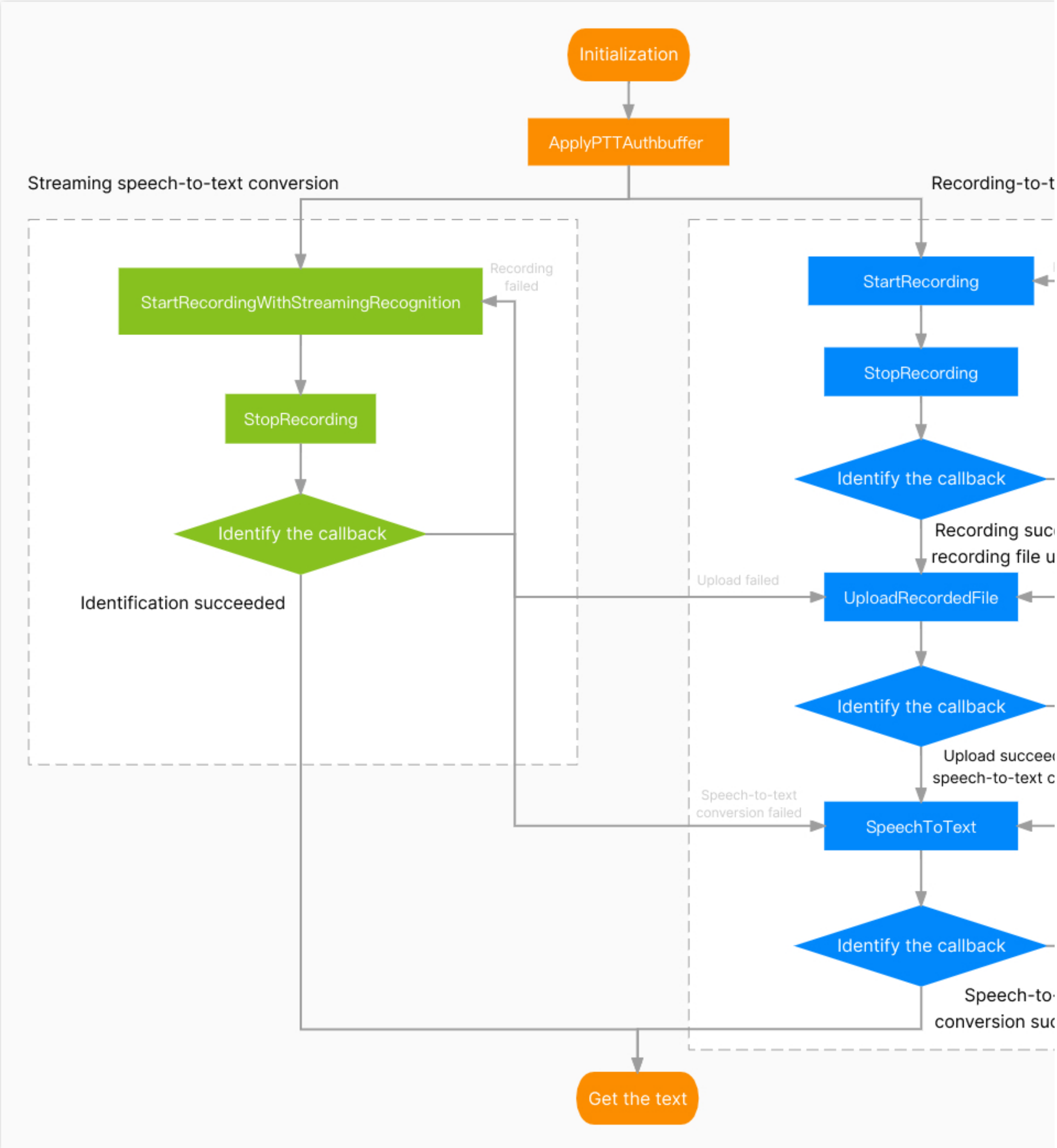
You do not need to enter a voice chat room when using the voice messaging service.

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to modify the maximum recording duration to 10 seconds, call the `SetMaxMessageLength` API to set it after initialization.

Flowchart for using the voice message service



Flowchart for using the speech-to-text service



API	Description
GenAuthBuffer	Gets the authentication information
SetMaxMessageLength	Specifies the maximum length of voice message

Generating the local authentication key

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.

API prototype

```
GenAuthBuffer(appId: string,roomId: string, openId:string, appKey: string) :string
```

Parameter	Type	Description
appId	string	`AppId` from the Tencent Cloud console.
roomId	string	Enter `null` or an empty string
openId	string	User ID, which is the same as `OpenId` during initialization.
key	string	Permission key from the Tencent Cloud console.

Application authentication

After the authentication information is generated, the authentication is assigned to the SDK.

API prototype

```
ApplyPTTAuthbuffer(authBuffer: string) :number
```

Parameter	Type	Description
authBuffer	string	Authentication information

Sample code

```
var authBuffer = m_context.GetAuthBuffer(UserConfig.GetAppID(), UserConfig.GetUserI
m_context.ApplyPTTAuthbuffer(authBuffer);
```

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

API prototype

```
PttSetMaxMessageLength(msTime: number) :number
```

Parameter	Type	Description

msTime	number	Audio duration in ms. Value range: 1000 < `msTime` <= 58000

Sample code

```
m_context.PttSetMaxMessageLength(58000);
```

Streaming Speech Recognition

Voice messaging and speech-to-text APIs

API	Description
PttStartRecordingWithStreamingRecognition	Starts streaming recording
PttStopRecording	Stops recording

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation. **To stop recording, call** Stop recording.

API prototype

```
PttStartRecordingWithStreamingRecognition(filePath: string, speechLanguage: string,
```

Parameter	Type	Description
filePath	string	Path of stored audio file
speechLanguage	string	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translateLanguage	string	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
string filePath = "xx/xxx/xxx.silk"
var ret = m_context.StartRecordingWithStreamingRecognition(filePath,"cmn-Hans-CN","
if (ret == 0) {
    this.currentStatus = "Start streaming recording";
```

```
} else {
    this.currentStatus = "Failed to start streaming recording";
}
```

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen on callback messages in the `OnEvent` notification, which is as detailed below:

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real time during the recording, which is equivalent to returning the recognized text while speaking.

The event message will be identified in the callback notification based on the actual needs. The passed parameters include the following four messages.

Message Name	Description
result	Return code indicating whether streaming speech recognition is successful
text	Text converted from speech
file_path	Local path of stored recording file
file_id	Backend URL address of recording file, which will be retained for 90 days

Note:

The file_id is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRecognition_IS_RUNNING' message is listened.

Error codes

Error Code	Description	Suggested Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.

32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

If the error code 4098 is reported, see Speech-to-text Conversion for solutions.

Sample code

```
m_context.setTMGDelegate(function(eventId, msg){
   switch (eventType) {
                    case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE:
                    {
                        HandleSTREAM2TEXTComplete(data,true);
                        break;
                        }
                    ...
                            case ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_R
                    {
                        HandleSTREAM2TEXTComplete(data, false);
                        break;
                    }
                }
});
```

Voice Message Recording

The recording process is as follows: start recording > stop recording > return recording callback > start the next recording.

Voice messaging and speech-to-text APIs

API	Description
PttStartRecording	Starts recording

PttPauseRecording	Pauses recording
PttResumeRecording	Resumes recording
PttStopRecording	Stops recording
PttCancelRecording	Cancels recording

Starting recording

This API is used to start recording.

API prototype

```
PttStartRecording(filePath: string) : number;
```

Parameter	Type	Description
filePath	string	Path of stored audio file

Sample code

```
string filepath = "xxxx/xxx.silk";
var ret = m_context.PttStartRecording(filepath);
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

API prototype

```
PttStopRecording() :number;
```

Sample code

```
m_context.PttStopRecording();
```

Callback for recording start

A callback will be executed through a delegate function to pass a message when recording is completed.
To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.

Parameter	Type	Description
code	string	0: Recording is completed
filepath	string	Path of stored recording file, which must be accessible and cannot be the `fileid`

Error codes

Error Code Value	Cause	Suggested Solution
4097	Parameter is empty.	Check whether the API parameters in the code are correct.
4098	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
4099	Recording is in progress.	Ensure that the SDK recording feature is used at the right time.
4100	Audio data is not captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	Mic permission is required for using the SDK. To add the permission, see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
m_context.setTMGDelegate(function(eventId, msg){
   switch (eventType) {
                    case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
                    {
                    // Process
                    break;
                        }
                    ...
```

```
                                case ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
                        {
                        // Process
                        break;
                        }
                }
});
```

Pausing recording

This API is used to pause recording. If you want to resume recording, call the `PttResumeRecording` API.

API prototype

```
PttPauseRecording() : number
```

Sample code

```
number ret = m_context.PttPauseRecording();
```

Resuming recording

This API is used to resume recording.

API prototype

```
PttResumeRecording() : number;
```

Sample code

```
number ret = m_context.PttResumeRecording();
```

Canceling recording

This API is used to cancel recording. **There is no callback after cancellation**.

API prototype

```
PttCancelRecording() : number
```

Sample code

```
m_context.PttCancelRecording();
```

Voice Message Upload, Download, and Playback

API	Description
PttUploadRecordedFile	Uploads an audio file
PttDownloadRecordedFile	Downloads an audio file
PttPlayRecordedFile	Plays back an audio file
PttStopPlayFile	Stops playing back an audio file
PttGetFileSize	Gets the audio file size
PttGetVoiceFileDuration	Gets the audio file duration

Uploading an audio file

This API is used to upload an audio file.

API prototype

```
PttUploadRecordedFile(filePath: string) : number
```

Parameter	Type	Description
filePath	String	Path of uploaded audio file, which is a local path.

Sample code

```
var ret = m_context.PttUploadRecordedFile(filePath);
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.
The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description

result	number	`0` : Recording is completed
filepath	string	Path of stored recording file
fileid	string	File URL

Error codes

Error Code Value	Cause	Suggested Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.
8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
8195	A network error occurred.	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device can access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
m_context.setTMGDelegate(function(eventId, msg){
 switch (eventType) {
           case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
           {
           // Process
           break;
               }
           ...
                   case ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
           {
```

```
                // Process
                break;
                }
            }
});
```

Downloading the audio file

This API is used to download an audio file.

API prototype

```
PttDownloadRecordedFile(fileId: string, filePath: string) : number
```

Parameter	Type	Description
fileId	string	File URL
filePath	string	Local path of saved file, which must be accessible and cannot be the `fileid`

Sample code

```
var ret = m_context.PttDownloadRecordedFile(fileID,filePath);
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	number	`0` : Download is completed
filepath	string	Path of stored recording file
fileid	string	URL of recording file, which will be retained on the server for 90 days.

Error codes

Error Code Value	Cause	Suggested Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.

12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.
12291	Network storage system exception.	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	Server file system error.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed during the process of getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.
12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.
12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice message and speech-to-text feature is initialized.

Sample code

```
m_context.setTMGDelegate(function(eventId, msg){
 switch (eventType) {
           case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
             }
        }
});
```

Playing back audio

This API is used to play back audio.

API prototype

```
PttPlayRecordedFile(filePath: string, voiceType: ITMG_VOICE_TYPE) : number
```

Parameter	Type	Description
filePath	string	Local audio file path
voicetype	ITMG_VOICE_TYPE	Voice changer type. For more information, see Voice Changing.

Error codes

Error Code Value	Cause	Suggested Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
m_context.PlayRecordedFile(filePath);
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameter includes `result` and `file_path` .

Parameter	Type	Description
code	number	`0` : Playback is completed
filepath	string	Path of stored recording file

Error codes

Error Code Value	Cause	Suggested Solution
20481	Initialization error.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	Parameter is empty.	Check whether the API parameters in the code

		are correct.
20484	Internal error.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
m_context.setTMGDelegate(function(eventId, msg){
 switch (eventType) {
        case ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE:
        {
        // Process
        break;
         }
        }
});
```

Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

API prototype

```
PttStopPlayFile() : number
```

Sample code

```
m_context.PttStopPlayFile();
```

Getting audio file size

This API is used to get the size of an audio file.

API prototype

```
PttGetFileSize(filePath: string) : number
```

Parameter	Type	Description
filePath	string	Path of audio file, which is a local path

Sample code

```
m_context.PttGetFileSize(filePath);
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

API prototype

```
PttGetVoiceFileDuration(filePath: string) : number
```

Parameter	Type	Description
filePath	string	Path of audio file, which is a local path

Sample code

```
number fileDuration = m_context.PttGetVoiceFileDuration(filePath);
```

Fast Recording-to-Text Conversion

Translating audio file into text in specified language

This API can specify a language for recognition or translate the information recognized in speech into a specified language and return the translation.

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

API prototype

```
PttSpeechToText(fileID: string, speechLanguage: string, translateLanguage: string)
```

Parameter	Type	Description
fileID	string	URL of audio file, which will be retained on the server for 90 days
speechLanguage	string	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translatelanguage	string	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
m_context.PttSpeechToText(filePath,"cmn-Hans-CN","cmn-Hans-CN");
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` and `text` (recognized text).

Parameter	Type	Description
result	number	`0` : Recording is completed
fileid	string	URL of recording file, which will be retained on the server for 90 days
text	string	Converted text

Error codes

Error Code Value	Cause	Suggested Solution
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	Network failed.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
32776	`authbuffer` check failed.	Check whether `authbuffer` is correct.
32784	Incorrect speech-to-text conversion parameter.	Check whether the API parameter `fileid` in the code is empty.
32785	Speech-to-text translation returned an error.	Error with the backend of voice messaging and speech-to-text feature. Analyze logs, get the actual

		error code returned from the backend to the client, and ask backend personnel for assistance.
32787	Speech-to-text conversion succeeded, but the text translation service was not activated.	Activate the text translation service in the console.
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

Sample code

```
m_context.setTMGDelegate(function(eventId, msg){
 switch (eventType) {
         case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
            {
            // Process
            break;
                }
            ...
                    case ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
            {
            // Process
            break;
            }
});
```

Voice Message Volume Level APIs

API	Description
PttGetMicLevel	Gets real-time mic volume level
PttSetMicVolume	Sets recording volume level
PttGetMicVolume	Gets recording volume level
PttGetSpeakerLevel	Gets real-time speaker volume level
PttSetSpeakerVolume	Sets playback volume
PttGetSpeakerVolume	Gets playback volume level

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. A number-type value will be returned. Value range: 0–200.

API prototype

```
PttGetMicLevel():number
```

Sample code

```
m_context.PttGetMicLevel();
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

API prototype

```
PttSetMicVolume(vol:number) :number
```

Parameter	Type	Description
vol	number	Value range: 0–200. Default value: `100`. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
m_context.PttSetMicVolume(vol);
```

Getting the recording volume of voice message

This API is used to get the recording volume of voice message. A number-type value will be returned. Value range: 0–200.

API prototype

```
PttGetMicVolume() : number
```

Sample code

```
m_context.PttGetMicVolume();
```


Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. A number-type value will be returned. Value range: 0-200.

API prototype

```
PttGetSpeakerLevel() : number;
```

Sample code

```
m_context.PttGetSpeakerLevel();
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice message. Value range: 0-200.

API prototype

```
PttSetSpeakerVolume(vol: number) : number
```

Sample code

```
m_context.PttSetSpeakerVolume(100);
```

Getting the playback volume of voice message

This API is used to get the playback volume of voice message. A number-type value will be returned. Value range: 0-200.

API prototype

```
PttGetSpeakerVolume() : number
```

Sample code

```
m_context.PttGetSpeakerVolume();
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for analysis.

API prototype

```
GetSDKVersion() :string
```

Sample code

```
string sdkVersion = m_context.GetSDKVersion();
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
SetLogLevel(level: number) : number
```

Parameter description

Parameter	Type	Description
level	ITMG_LOG_LEVEL	Sets the log level. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO` .

`level` description:

Value of `level`	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
m_context.SetLogLevel(TMG_LOG_LEVEL_INFO);
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

OS	Path
Windows	%appdata%\\GMEGLOBAL\\GME\\ProcessName

API prototype

```
SetLogPath(logPath: string)
```

Parameter	Type	Description
logPath	string	Path

Sample code

```
string logDir = ""// Set a path by yourself
m_context.SetLogPath(logDir);
```

SDK for Flutter

Integrating SDK

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document describes how to configure a Flutter project for the GME APIs for Flutter.

Supported Platforms

The GME SDK for Flutter supports iOS and Android platforms.

Importing the SDK

Step 1. Download the GME SDK for Flutter

Download the SDK file in SDK Download Guide, which contains the GME plugin. Decompress the SDK file to a local directory.

Step 2. Add dependencies of the GME plugin to the Flutter project

Add GME dependencies to the `pubspec.yaml` file in your Flutter project. Note that the `path` parameter is the path where the SDK file is decompressed to.

```
dependencies:
  flutter:
    sdk: flutter
  gme:
    path: ../
```

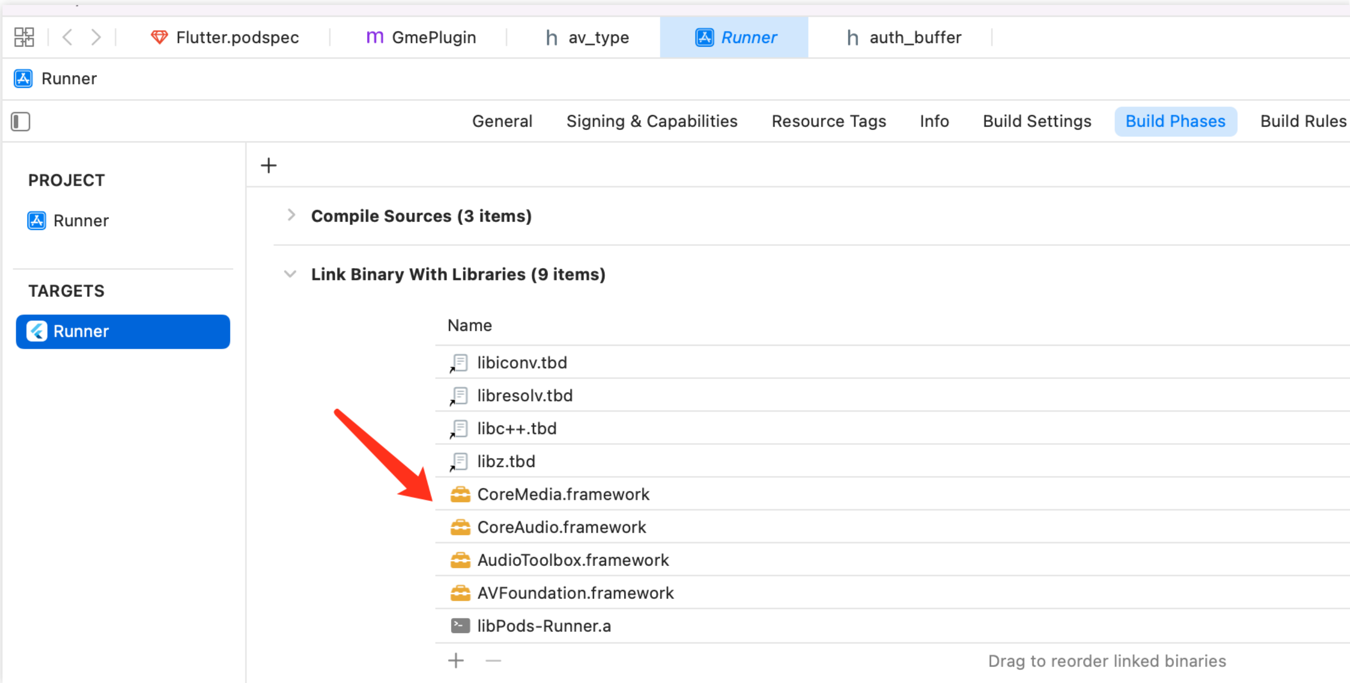
After saving the `pubspec.yaml` file, run the `flutter pub get` command on the CLI to make the GME plugin in the project take effect (if the Flutter plugin is configured in your IDE, this command will be executed automatically once the file is saved).

```
flutter pub get
```

Modifying the iOS Project

1. In Terminal, go to the iOS project directory of your Flutter project and run `pod install`.

2. In the Xcode project, configure the following GME dependent library files (you can skip this step if such dependencies already exist in your project):



3. The GME SDK for iOS requires the following permissions:
Required background modes: Allows running in the background (optional).
Microphone Usage Description: Allows access to microphone.

Modifying the Android Project

1. As GME requires permissions such as call permission and uses the permission management plugin `flutter permission-handler` , you need to modify the project as follows to use the SDK for Android 31 or later (skip this step if the SDK is already in the project):

```
android {
    compileSdkVersion 31

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }

    kotlinOptions {
        jvmTarget = '1.8'
    }

    sourceSets {
        main.java.srcDirs += 'src/main/kotlin'
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID (https://developer.android.com/studio
        applicationId "com.example.flutter_android"
        minSdkVersion 16
        targetSdkVersion 31
        versionCode flutterVersionCode.toInteger()
        versionName flutterVersionName
    }

    buildTypes {
        release {
```

2. Add the project permissions to the Flutter project file `android/app/src/AndroidManifest.xml` (skip this step if such permissions have been added):

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```


Real-time Voice

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document describes how to integrate with and debug GME client APIs for the real-time voice chat feature for Flutter.

Key Considerations for Using GME

GME provides the real-time voice chat service, voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

Notes

You have created a GME application and obtained the SDK `AppID` and key. For more information, see Activating Services.

You have activated **GME real-time voice chat, voice messaging and speech-to-text services**. For more information, see Activating Services.

Configure your project before using GME; otherwise, the SDK will not take effect.

After a GME API is called successfully, `GmeError.AV_OK` will be returned with the value being `0`.

GME APIs should be called in the same thread.

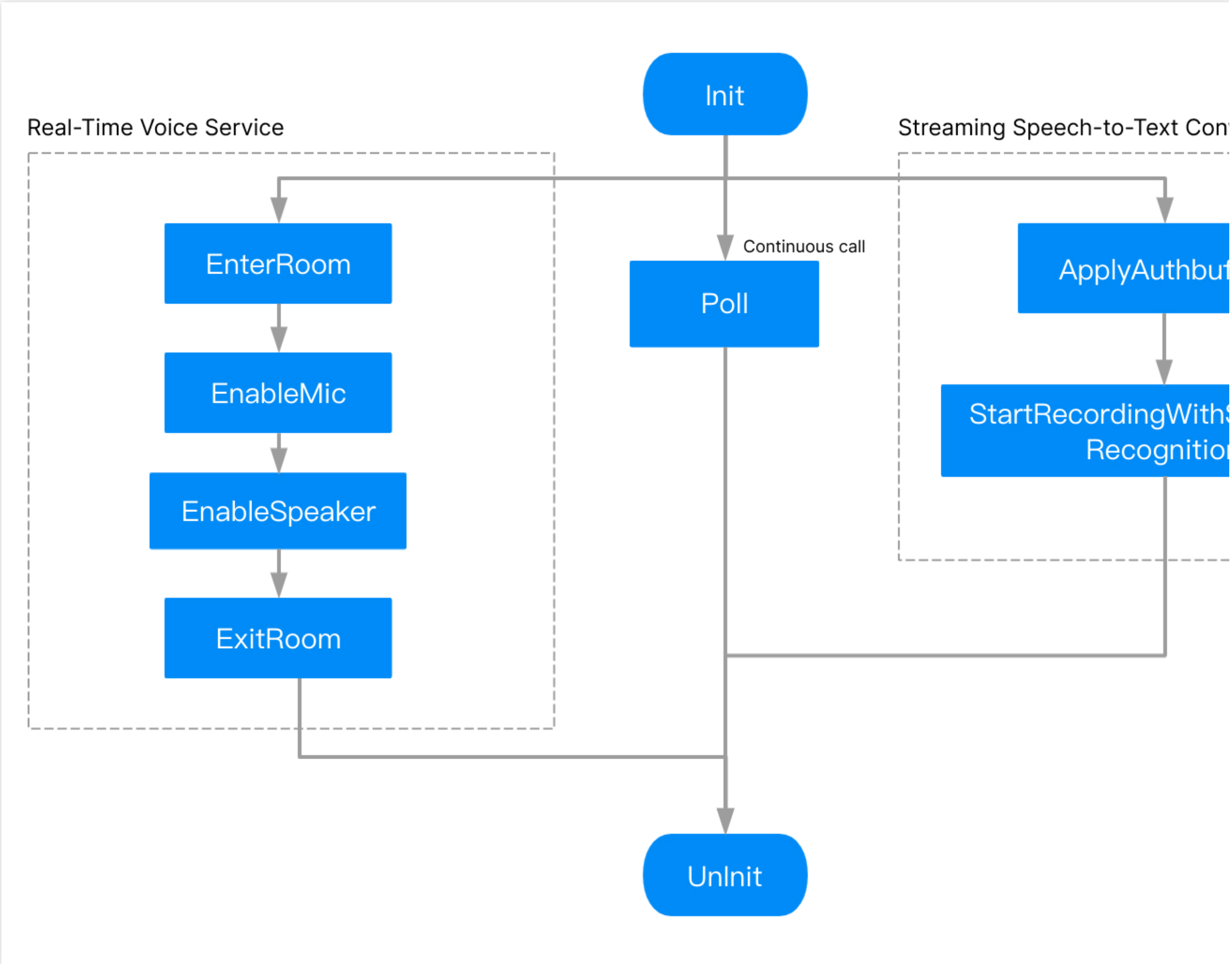
The `Poll` API should be called periodically for GME to trigger event callbacks.

For detailed error codes, see Error Codes.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initialize GME.
2. Call `Poll` periodically to trigger event callbacks.
3. Enter a voice chat room.
4. Turn on the mic.
5. Turn on the speaker.
6. Exit the voice chat room.
7. Uninitialize GME.

Core APIs

API	Description
Init	Initializes GME.
Poll	Triggers the event callback.

Pause	Pauses the system.
Resume	Resumes the system.
Uninit	Uninitializes GME.

Importing the GME module

```
import 'package:gme/gme.dart';
import 'package:gme/gmeType.dart';
```

Getting an instance

To use the voice chat feature, get the `GmeSDK` object first.

```
ITMGContext context = ITMGContext.GetInstance();
```

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice, voice message, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
//class ITMGContext
Future<int> InitSDK(String appID, String openID)
```

Parameter	Type	Description
sdkAppId	string	`AppID` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	string	`openID` can only be in `int64` type, which is passed in after being converted to a string. You can customize its rules, and it must be unique in the application. To pass in `openID` as a string, submit a ticket for application.

Returned values

Returned Value	Description
GmeError.AV_OK= 0	SDK initialized successfully.

AV_ERR_SDK_NOT_FULL_UPDATE=7015	Solution: Check whether the SDK file is complete. We recommend that you delete it and then import the SDK again.

Note:

Notes on 7015 error code

The 7015 error code is identified by MD5. If this error is reported during integration, check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Due to the third-party enhancement, the Unity packaging mechanism, and other factors, the MD5 value of the library file will be affected, resulting in misjudgment. Therefore, **ignore this error in the logic for official releases**, and avoid displaying it on the UI.

Sample code

```
string SDKAPPID3RD = "14000xxxxx";
string openId="10001";
int res = await ITMGContext.GetInstance().InitSDK(SDKAPPID3RD, openId);

if (ret != GmeError.AV_OK)
{
    print("Init SDK Error");
    return;
}
```

Setting callbacks

The API class uses the `Delegate` method to send callback notifications to the application. Register the callback function to the SDK for receiving callback messages before room entry.

Function prototype and sample code

Register the callback function to the SDK for receiving callback messages before room entry.

```
// When initializing the SDK
ITMGContext.GetInstance().SetEvent(handleEventMsg);
// Callback method
void handleEventMsg(int eventType, String data) async {
    // enterRoom event
    print("AddDelegate3" + eventType.toString());
    switch (eventType) {
      case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
        // Callback of room entry
```

```
            }
            break;
          case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE:
            {
             // Callback of room switch
            }
            break;
        }
    }
```

Triggering event callback

You need to periodically call the `Poll` API to trigger event callbacks. The `Poll` API is GME's message pump and should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run abnormally. For more information, see the `EnginePollHelper` file in SDK Download Guide.

Note:

Call the `Poll` API periodically

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
Future<void> Poll();
```

Sample code

```
  Future<void> pollTimer() async {_pollTimer = Timer.periodic(Duration(milliseconds
      ITMGContext.GetInstance().Poll();
    });
  }
```

Pausing the system

When a `Pause` event occurs in the system, the engine should also be notified for pause. For example, when the application switches to the background (OnApplicationPause, isPause=True), and you do not need the background to play back the audio in the room, please call `Pause` API to pause the GME service.

API prototype

```
Future<int> Pause()
```

Resuming the system

When a `Resume` event occurs in the system, the engine should also be notified for resumption. The `Resume` API only supports resuming voice chat.

API prototype

```
Future<int> Resume()
```

Uninitializing SDK

This API is used to uninitialize the SDK. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

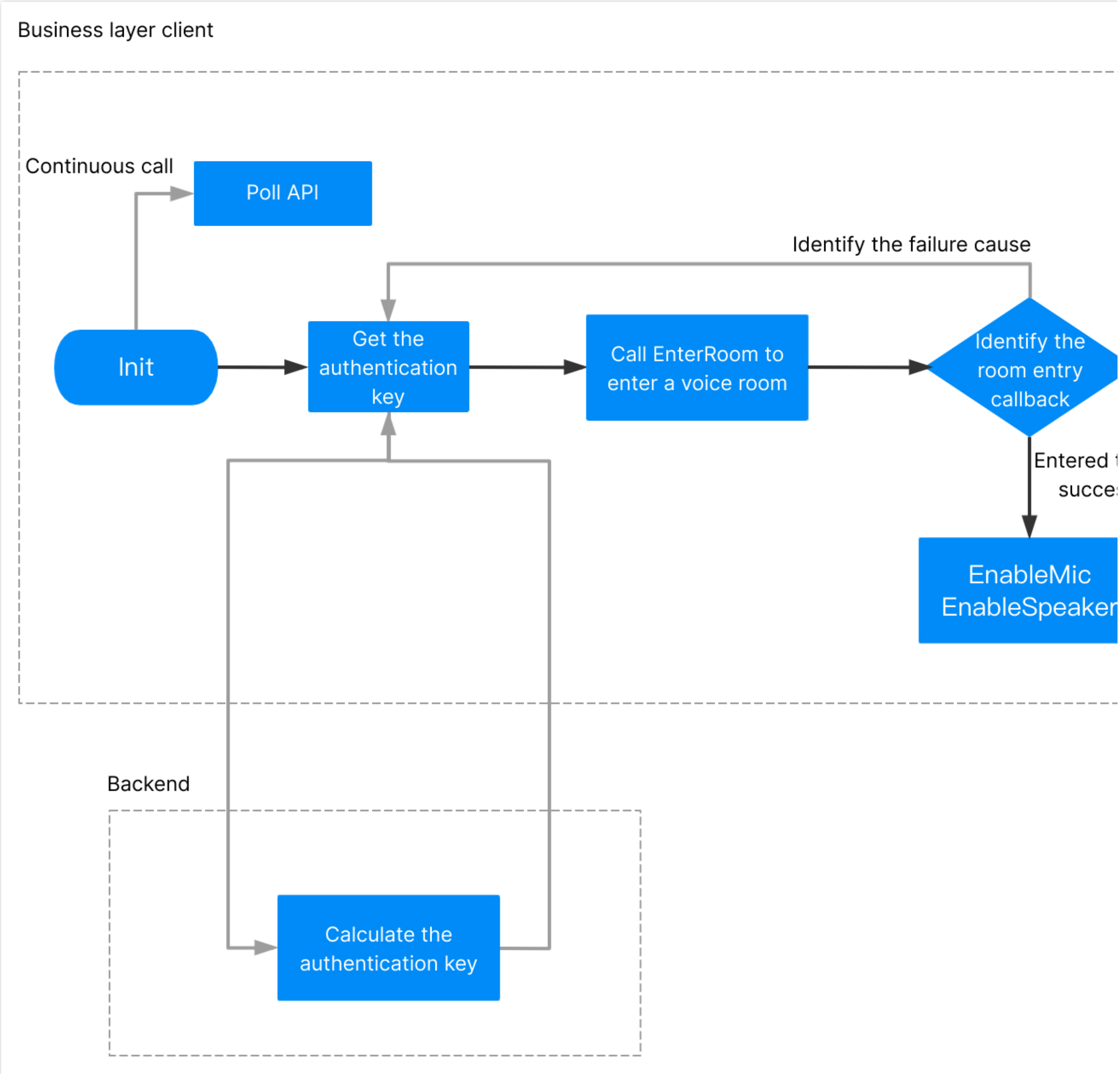
API prototype

```
Future<int> Uninit()
```

Voice Chat Room APIs

You should initialize and call the SDK to enter a room before voice chat can start.

If you have any questions when using the service, see Sound and Audio.



API	Description
GenAuthBuffer	Calculates the local authentication key.
EnterRoom	Enters a room.
ExitRoom	Exits a room.
IsRoomEntered	Determines whether room entry is successful.

Local authentication key calculation

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.

API prototype

```
Future<Uint8List> GenAuthBuffer(String appID, String roomID, String openID, String
```

Parameter	Type	Description
appID	string	`AppID` from the Tencent Cloud console
roomID	string	Room ID, which can contain up to 127 characters.
openID	string	User ID, which is the same as `openID` during initialization.
key	string	Permission key from the Tencent Cloud console.

Sample code

```
Uint8List userSig = await ITMGContext.GetInstance().GenAuthBuffer(_editAppID.text,
int res = await ITMGContext.GetInstance().EnterRoom(_editRoomID.text, 1, userSig);
```

Entering a room

This API is used to enter a room with the generated authentication information. The mic and speaker are not enabled by default after room entry.

Note:

If the room entry callback result is `0`, the room entry is successful. If `0` is returned from the `EnterRoom` API, it doesn't necessarily mean that the room entry is successful.

The audio type of the room is determined by the first user entering the room. After that, if a member in the room changes the room type, it will take effect for all members there. For example, if the first user entering the room uses the smooth sound quality, and the second user entering the room uses the HD sound quality, the room audio type of the second user will change to the smooth sound quality. Only after a member in the room calls the `ChangeRoomType` API will the audio type of the room be changed.

API prototype

```
Future<int> EnterRoom(String roomID, int roomType, Uint8List authBuffer)
```

Parameter	Type	Description
roomId	string	Room ID, which can contain up to 127 characters.

roomType	ITMGRoomType	Room type. We recommend that you select `ITMG_ROOM_TYPE_FLUENCY` for games. For more information on room audio types, see Sound Quality.
appKey	Uint8List	Authentication key

Sample code

```
int res = await ITMGContext.GetInstance().EnterRoom(_editRoomID.text, 1, authBuffer
```

Callback for room entry

After the user enters the room, the `ITMG_MAIN_EVENT_TYPE_ENTER_ROOM` event type will be called back to notify the room entry result, which can be listened on for processing. A successful callback means that the room entry is successful, and the billing **starts**.

Note:

Billing references:

Purchase Guide

Billing

Will the billing continue if the client is disconnected from the server when using the voice chat?

Sample code

```
// Listen on an event:
void handleEventMsg(int eventType, String data) async {
  switch (eventType) {
      case ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
      {
        // The process after room entry
      }
  }
}
ITMGContext.GetInstance().SetEvent(handleEventMsg);
```

Data details

Message	Data	Sample
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	result; error_info	{"error_info":"","result":0}
ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	result; error_info	{"error_info":"waiting timeout, please check your network","result":0}

If the network is disconnected, there will be a disconnection callback notification `ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT` . At this time, the SDK will automatically reconnect, and the callback is `ITMG_MAIN_EVENT_TYPE_RECONNECT_START` . When the reconnection is successful, there will be a callback `ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS` .

Error codes

Error Code	Cause and Solution
7006	Authentication failed. Possible causes: - The `AppID` does not exist or is incorrect.- An error occurred while authenticating the `authbuff`.- Authentication expired.- The `OpenId` does not meet the specification.
7007	The user was already in another room.
1001	The user was already in the process of entering a room but repeated this operation. We recommend that you not call the room entering API until the room entry callback is returned.
1003	The user was already in the room and called the room entering API again.
1101	Make sure that the SDK is initialized, `OpenId` complies with the rules, the APIs are called in the same thread, and the `Poll` API is called normally.

Exiting a room

This API is used to exit the current room. It is an async API. The returned value `AV_OK` indicates a successful async delivery. If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API; instead, you can directly call the `EnterRoom` API.

API prototype

```
Future<int> ExitRoom()
```

Sample code

```
ITMGContext.GetInstance().ExitRoom();
```

Callback for room exit

After the user exits a room, a callback will be returned with the message being `ITMG_MAIN_EVENT_TYPE_EXIT_ROOM` . The sample code is shown below:

Sample code


```
void handleEventMsg(int eventType, String data) async{
  switch (eventType) {
      case ITMG_MAIN_EVENT_TYPE_EXIT_ROOM:
      {
      // The process after room exit
        break;
      }
  }
}
ITMGContext.GetInstance().SetEvent(handleEventMsg);
```

Determining whether user has entered room

This API is used to determine whether the user has entered a room. A value in bool type will be returned. Do not call this API during room entry.

API prototype

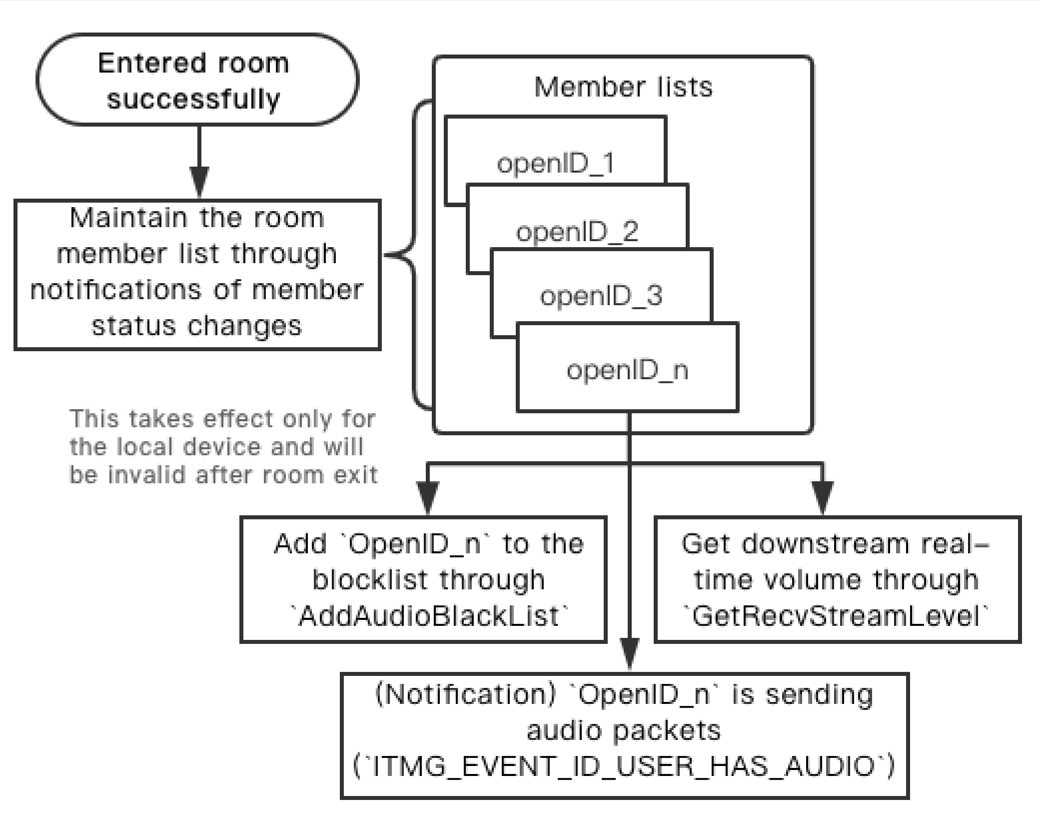
```
Future<bool> IsRoomEntered()
```

Sample code

```
bool res = await ITMGContext.GetInstance().IsRoomEntered();
```

Room Status Maintenance

APIs in this section are used to display speaking members and members entering or exiting the room and mute a member in the room at the business layer.



API/Notification	Description
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	The member status changed.
AddAudioBlackList	Mutes a member in the room.
RemoveAudioBlackList	Unmutes a user.
IsOpenIdInAudioBlackList	Queries whether the user of the specified `openid` is muted.

Notification events of member room entry and speaking status

This event is used to get speaking users in the room and display the users on the UI, and to send a notification when someone enters or exits the room.

A notification for this event will be sent only when the status changes. To get the member status in real time, cache the notification when it is received at the business layer. The event message `ITMG_MAIN_EVNET_TYPE_USER_UPDATE` containing `event_id`, `count`, and `openIdList` will be returned, which will be identified in the `OnEvent` notification.

Notifications of the `EVENT_ID_ENDPOINT_NO_AUDIO` audio event will be sent only when the threshold is exceeded, that is, other members in the room can receive the notification that the local user stops speaking only after the local client captures no voice for two seconds.

The audio event returns only the member speaking status but not the specific volume level. If you need the specific volume levels of members in the room, you can use the `GetVolumeById` API.

event_id	Description	Maintenance
EVENT_ID_ENDPOINT_ENTER	Return the `openid` of the member entering the room.	Member list
EVENT_ID_ENDPOINT_EXIT	Return the `openid` of the member exiting the room.	Member list
EVENT_ID_ENDPOINT_HAS_AUDIO	Return the `openid` of the member sending audio packets in the room. This event can be used to determine whether a user is speaking and display the voiceprint effect.	Chat member list
EVENT_ID_ENDPOINT_NO_AUDIO	Return the `openid` of the member stopping sending audio packets in the room.	Chat member list

Sample code

```
void handleEventMsg(int eventType, String data) async {
  if (eventType == ITMG_MAIN_EVENT_TYPE_ENTER_ROOM)
  {
              // Process
            switch (eventID)
             {
             case EVENT_ID_ENDPOINT_ENTER:
                  // A member enters the room
                  break;
             case EVENT_ID_ENDPOINT_EXIT:
                  // A member exits the room
                break;
            case EVENT_ID_ENDPOINT_HAS_AUDIO:
                // A member sends audio packets
                break;
            case EVENT_ID_ENDPOINT_NO_AUDIO:
                // A member stops sending audio packets
                break;

            default:
                break;
             }
        break;
  }
}
```

Muting a member in the room

This API is used to add an ID to the audio data blocklist. This operation blocks audio from someone and only applies to the local device without affecting other devices. The returned value `0` indicates that the call is successful.

Assume that users A, B, and C are all speaking using their mic in a room:

If A blocks C, A can only hear B;

If B blocks neither A nor C, B can hear both of them;

If C blocks neither A nor B, C can hear both of them.

This API is suitable for scenarios where a user is muted in a room.

API prototype

```
Future<int> AddAudioBlackList(String openID)
```

Parameter	Type	Description
openID	string	`openid` of the user to be blocked

Sample code

```
res = await ITMGContext.GetInstance().GetAudioCtrl().AddAudioBlackList(_editRoomMan
```

Unmuting

This API is used to remove an ID from the audio data blocklist. A returned value of 0 indicates the call is successful.

API prototype

```
Future<int> RemoveAudioBlackList(String openID)
```

Parameter	Type	Description
openId	string	ID to be unblocked

Sample code

```
res = await ITMGContext.GetInstance().GetAudioCtrl().RemoveAudioBlackList(_editRoom
```

Voice Chat Capturing APIs

The voice chat APIs can only be called after SDK initialization and room entry.
When the user clicks the button of enabling/disabling the mic or speaker on the UI, we recommend you call the `EnableMic` or `EnableSpeaker` API.
To enable the user to press the mic button on the UI to speak and release it to stop speaking, we recommend you call `EnableAudioCaptureDevice` once during room entry and call `EnableAudioSend` to enable the user to speak while pressing the button.

API	Description
EnableMic	Enables/Disables the mic.
GetMicState	Gets the mic status.
EnableAudioCaptureDevice	Enables/Disables the capturing device.
IsAudioCaptureDeviceEnabled	Gets the capturing device status.
EnableAudioSend	Enables/Disables audio upstreaming.
IsAudioSendEnabled	Gets the audio upstreaming status.
GetMicLevel	Gets the real-time mic volume level.
GetSendStreamLevel	Gets the real-time audio upstreaming volume level.
SetMicVolume	Sets the mic volume level.
GetMicVolume	Gets the mic volume level.

Enabling or disabling mic

This API is used to enable/disable the mic. The mic and speaker are not enabled by default after room entry.
EnableMic = EnableAudioCaptureDevice + EnableAudioSend

API prototype

```
Future<int> EnableMic(bool enable)
```

Parameter	Type	Description
isEnabled	bool	To enable the mic, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
// Turn on mic
int res = await ITMGContext.GetInstance().GetAudioCtrl().EnableMic(true);
```

Getting the mic status

This API is used to get the mic status. The returned value 0 indicates that the mic is off, while 1 is on.

API prototype

```
Future<int> GetMicState()
```

Sample code

```
int micState = await ITMGContext.GetInstance().GetAudioCtrl().GetMicState();
```

Enabling or disabling capturing device

This API is used to enable/disable a capturing device. The device is not enabled by default after room entry.

This API can only be called after room entry. The device will be disabled automatically after room exit.

Operations such as permission application and volume type adjustment will generally be performed when a capturing device is enabled on a mobile device.

API prototype

```
Future<int> EnableAudioCaptureDevice(bool enable)
```

Parameter	Type	Description
enable	bool	To enable the capturing device, set this parameter to `true` , otherwise, set it to `false` .

Sample code

```
// Enable capturing device
int res = await ITMGContext.GetInstance().GetAudioCtrl().EnableAudioCaptureDevice(t
```

Getting the capturing device status

This API is used to get the status of a capturing device.

API prototype


```
Future<bool> IsAudioCaptureDeviceEnabled()
```

Sample code

```
bool res = await ITMGContext.GetInstance().GetAudioCtrl().IsAudioCaptureDeviceEnabl
```

Enabling or disabling audio upstreaming

This API is used to enable/disable audio upstreaming. If a capturing device is already enabled, it will send captured audio data; otherwise, it will remain mute. For more information on how to enable/disable the capturing device, see the `EnableAudioCaptureDevice` API.

API prototype

```
Future<int> EnableAudioSend(bool enable)
```

Parameter	Type	Description
isEnabled	bool	To enable audio upstreaming, set this parameter to `true` ; otherwise, set it to `false` .

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().EnableAudioSend(isCheck);
```

Getting audio upstreaming status

This API is used to get the status of audio upstreaming.

API prototype

```
Future<bool> IsAudioSendEnabled()
```

Sample code

```
bool IsAudioSend = await ITMGContext.GetInstance().GetAudioCtrl().IsAudioSendEnable
```

Getting the real-time mic volume

This API is used to get the real-time mic volume level. A number-type value in the range of 0–100 will be returned. We recommend that you call this API once every 20 ms.

API prototype

```
Future<int> GetMicLevel()
```

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().GetMicLevel();
```

Getting the real-time audio upstreaming volume

This API is used to get the local real-time audio upstreaming volume level. A number-type value in the range of 0–100 will be returned.

API prototype

```
Future<int> GetSendStreamLevel()
```

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().GetSendStreamLevel();
```

Setting the mic software volume

This API is used to set the mic volume level. The corresponding parameter is `volume`, which is equivalent to attenuating or gaining the captured sound.

API prototype

```
Future<int> SetMicVolume(int volume)
```

Parameter	Type	Description
volume	number	Value range: 0–200. Default value: `100`. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
int volume = 100;
int res = await ITMGContext.GetInstance().GetAudioCtrl().SetMicVolume(volume);
```

Getting the mic software volume

This API is used to get the mic volume level. A number-type value will be returned. `101` indicates that the `SetMicVolume` API has not been called.

API prototype

```
Future<int> GetMicVolume()
```

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().GetMicVolume();
```

Voice Chat Playback APIs

API	Description
EnableSpeaker	Enables/Disables the speaker.
GetSpeakerState	Gets the speaker status.
EnableAudioPlayDevice	Enables/Disables the playback device.
IsAudioPlayDeviceEnabled	Gets the playback device status.
EnableAudioRecv	Enables/Disables audio downstreaming.
IsAudioRecvEnabled	Gets the audio downstreaming status.
GetSpeakerLevel	Gets the real-time speaker volume level.
GetRecvStreamLevel	Gets the real-time downstreaming audio levels of other members in the room.
SetSpeakerVolume	Sets the speaker volume level.
GetSpeakerVolume	Gets the speaker volume level.

Enabling or disabling speaker

This API is used to enable/disable the speaker. **EnableSpeaker = EnableAudioPlayDevice + EnableAudioRecv**

API prototype

```
Future<int> EnableSpeaker(bool enable)
```


Parameter	Type	Description
bEnable	bool	To disable the speaker, set this parameter to `false`; otherwise, set it to `true`.

Sample code

```
// Turn on the speaker
await ITMGContext.GetInstance().GetAudioCtrl().EnableSpeaker(isCheck);
```

Getting the speaker status

This API is used to get the speaker status. 0 indicates that the speaker is off, and 1 is on.

API prototype

```
Future<int> GetSpeakerState()
```

Sample code

```
int spkState = await ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerState();
```

Enabling or disabling playback device

This API is used to enable/disable a playback device.

API prototype

```
Future<int> EnableAudioPlayDevice(bool enable)
```

Parameter	Type	Description
enable	bool	To disable the playback device, set this parameter to `false`; otherwise, set it to `true`.

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().EnableAudioPlayDevice(isCh
```

Getting the playback device status

This API is used to get the status of a playback device.

API prototype

```
Future<bool> IsAudioPlayDeviceEnabled()
```

Sample code

```
bool res = await ITMGContext.GetInstance().GetAudioCtrl().IsAudioPlayDeviceEnabled(
```

Enabling or disabling audio downstreaming

This API is used to enable/disable audio downstreaming. If a playback device is already enabled, it will play back audio data from other members in the room; otherwise, it will remain mute. For more information on how to enable/disable the playback device, see the `EnableAudioPlayDevice` API.

API prototype

```
Future<int> EnableAudioRecv(bool enable)
```

Parameter	Type	Description
isEnabled	bool	To enable audio downstreaming, set this parameter to `true`; otherwise, set it to `false`.

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().EnableAudioRecv(isCheck);
```

Getting audio downstreaming status

This API is used to get the status of audio downstreaming.

API prototype

```
Future<bool> IsAudioRecvEnabled()
```

Sample code

```
bool res = await ITMGContext.GetInstance().GetAudioCtrl().IsAudioRecvEnabled();
```

Getting the real-time speaker volume

This API is used to get the real-time speaker volume level. A number-type value will be returned to indicate the volume level. We recommend that you call this API once every 20 ms.

API prototype

```
Future<int> GetSpeakerLevel()
```

Sample code

```
bool res = await ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerLevel();
```

Getting the real-time downstreaming audio levels of other members in room

This API is used to get the real-time audio downstreaming volume of other members in the room. A number-type value will be returned. Value range: 0–200.

API prototype

```
Future<int> GetRecvStreamLevel(String openID)
```

Parameter	Type	Description
openId	string	\`openId\` of another member in the room

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().GetRecvStreamLevel(_editRo
```

Dynamically setting the volume of a member of the room

This API is used to set the volume of a member in the room. It takes effect only on the local.

API prototype

```
Future<int> SetSpeakerVolumeByOpenID(String openId, int volume)
```

Parameter	Type	Description
openId	string	\`OpenID\` of the target user
volume	number	Percentage. Recommended value range: 0-200. Default value: \`100\`.

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().SetSpeakerVolumeByOpenID(_
```


Getting volume percentage

This API is used to get the volume level set by `SetSpeakerVolumeByOpenID` .

API prototype

```
Future<int> GetSpeakerVolumeByOpenID(String openId)
```

Parameter	Type	Description
openId	string	`OpenID` of the target user

Returned values

API returns volume percentage set by OpenID, where 100 is by default.

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerVolumeByOpenID(_
```

Setting the speaker volume

This API is used to set the speaker volume.

API prototype

```
Future<int> SetSpeakerVolume(int volume)
```

Parameter	Type	Description
volume	number	Volume level. Value range: 0-200. Default value: `100`. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
int volume = value.toInt();
int res = await ITMGContext.GetInstance().GetAudioCtrl().SetSpeakerVolume(volume);
```

Getting the speaker volume

This API is used to get the speaker volume. A number-type value will be returned to indicate the volume. `101` indicates that the `SetSpeakerVolume` API has not been called.
"Level" indicates the real-time volume, and "Volume" the speaker volume. The final volume = Level * Volume%. For example, if the "Level" is 100 and "Volume" is 60, the final volume is "60".

API prototype

```
Future<int> GetSpeakerVolume()
```

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().GetSpeakerVolume();
```

Advanced APIs

Enabling in-ear monitoring

This API is used to enable in-ear monitoring. You need to call `EnableLoopBack+EnableSpeaker` before you can hear your own voice.

API prototype

```
Future<int> EnableLoopBack(bool enable)
```

Parameter	Type	Description
enable	bool	Specifies whether to enable.

Sample code

```
int res = await ITMGContext.GetInstance().GetAudioCtrl().EnableLoopBack(true);
```

Getting user's room audio type

This API is used to get a user's room audio type. The returned value is the room audio type. Value 0 indicates that an error occurred while getting the user's room audio type. For room audio types, see the `EnterRoom` API.

API prototype

```
Future<int> GetRoomType()
```

Sample code

```
int curType = await ITMGContext.GetInstance().GetRoom().GetRoomType();
```

Changing the room type

This API is used to modify a user's room audio type. For the result, see the callback event. The event type is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` . The audio type of the room is determined by the first user to enter the room. After that, if a member in the room changes the room type, it will take effect for all members there.

API prototype

```
Future<int> ChangeRoomType(int roomType)
```

Parameter	Type	Description
roomtype	number	Room type to be switched to. For room audio types, see the `EnterRoom` API.

Sample code

```
int res = await ITMGContext.GetInstance().GetRoom().ChangeRoomType(1);
```

Callback event

After the room type is set, the event message `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE` will be returned in the callback. The returned parameters include `result` , `error_info` , and `new_room_type` . The `new_room_type` represents the following information. The event message will be identified in the `OnEvent` function.

Event Subtype	Parameter	Description
ITMG_ROOM_CHANGE_EVENT_ENTERROOM	1	Indicates that the existing audio type is inconsistent with and changed to that of the entered room.
ITMG_ROOM_CHANGE_EVENT_START	2	Indicates that a user is already in the room and the audio type starts changing (e.g., calling the `ChangeRoomType` API to change the audio type).
ITMG_ROOM_CHANGE_EVENT_COMPLETE	3	Indicates that a user is already in the room and the audio type has been changed.
ITMG_ROOM_CHANGE_EVENT_REQUEST	4	Indicates that a room member calls the `ChangeRoomType` API to request a change of room audio type.

Sample code

```
case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE:
  {
        // Process room type events
  }
  break;
```

The monitoring event of room call quality

This is the quality monitoring event used to listen on the network quality. If your network conditions are poor, the business layer will ask you to switch the network through the UI. This event is triggered once every two seconds after room entry, and its message is `ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY` . The returned parameters include `weight` , `loss` , and `delay` , which are as detailed below:

Parameter	Type	Description
weight	number	Value range: 1–50. \`50\` indicates excellent sound quality, \`1\` indicates very poor (barely usable) sound quality, and \`0\` represents an initial meaningless value. Generally, if the value is below 30, you can remind users that the network is poor and recommend them to switch the network.
loss	var	Upstream packet loss rate
delay	number	Voice chat delay in ms

Getting version number

This API is used to get the SDK version number for analysis.

API prototype

```
Future<String> GetSDKVersion()
```

Sample code

```
_sdkVersions = await ITMGContext.GetInstance().GetSDKVersion();
```

Setting the application name and version

This API is used to set the application name and version.

API prototype

```
Future<void> SetAppVersion(String appVersion)
```

Parameter description

Parameter	Type	Description
appVersion	string	Application name and version

Sample code

```
await ITMGContext.GetInstance().SetAppVersion("gme V2.0.0");
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
Future<int> SetLogLevel(int levelWrite, int levelPrint)
```

Parameter description

Parameter	Type	Description
level	number	Sets the log level. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO`.

`level` description:

level	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContext.GetInstance().SetLogLevel(ITMG_LOG_LEVEL.TMG_LOG_LEVEL_ERROR, ITMG_LOG_
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

API prototype

```
Future<int> SetLogPath(String logDir)
```

Parameter	Type	Description
logPath	string	Path

Sample code

```
String curPath = ""// Set a path by yourself
ITMGContext.GetInstance().SetLogPath(curPath);
```

Getting the diagnostic messages

This API is used to get information on the quality of real-time audio/video calls, which is mainly used to view real-time call quality and troubleshoot and can be ignored on the business side.

API prototype

```
Future<String> GetQualityTips()
```

Sample code

```
String curQualityTips = await ITMGContext.GetInstance().GetRoom().GetQualityTips();
```

Callback message

Message	Description	Data
ITMG_MAIN_EVENT_TYPE_ENTER_ROOM	A member entered the audio room.	result; error
ITMG_MAIN_EVENT_TYPE_EXIT_ROOM	A member exited the audio room.	result; error

ITMG_MAIN_EVENT_TYPE_ROOM_DISCONNECT	The room was disconnected for network or other reasons.	result; error
ITMG_MAIN_EVNET_TYPE_USER_UPDATE	Room members were updated.	user_list; event_id
ITMG_MAIN_EVENT_TYPE_RECONNECT_START	The reconnection to the room started.	result; error
ITMG_MAIN_EVENT_TYPE_RECONNECT_SUCCESS	The reconnection to the room succeeded.	result; error
ITMG_MAIN_EVENT_TYPE_SWITCH_ROOM	The room was quickly switched.	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_TYPE	The room status was changed.	result; error_info; sub_event_ new_room_
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_START	Cross-room mic connect started.	result;
ITMG_MAIN_EVENT_TYPE_ROOM_SHARING_STOP	Cross-room mic connect stopped.	result;
ITMG_MAIN_EVENT_TYPE_SPEAKER_DEFAULT_DEVICE_CHANGED	The default speaker device was changed.	result; error
ITMG_MAIN_EVENT_TYPE_SPEAKER_NEW_DEVICE	A new speaker	result; error

	device was added.	
ITMG_MAIN_EVENT_TYPE_SPEAKER_LOST_DEVICE	A speaker device was lost.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_NEW_DEVICE	A new mic device was added.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_LOST_DEVICE	A mic device was lost.	result; error
ITMG_MAIN_EVENT_TYPE_MIC_DEFAULT_DEVICE_CHANGED	The default mic device was changed.	result; error
ITMG_MAIN_EVENT_TYPE_CHANGE_ROOM_QUALITY	The room quality changed.	weight; loss delay
ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE	Voice message recording was completed.	result; file_p
ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE	Voice message upload was completed.	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE	Voice message download was completed.	result; file_path;file
ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE	Voice message playback was completed.	result; file_p

ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE	Fast speech-to-text conversion was completed.	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE	Streaming speech-to-text conversion was completed.	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING	Streaming speech-to-text conversion is in progress.	result; file_p text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TEXT2SPEECH_COMPLETE	Text-to-speech conversion was completed.	result; text;file_id
ITMG_MAIN_EVNET_TYPE_PTT_TRANSLATE_TEXT_COMPLETE	Text translation was completed.	result; text;file_id

Speech-to-Text Service

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document describes how to integrate with and debug GME client APIs for the voice messaging and speech-to-text services for Flutter.

Key Considerations for Using GME

GME provides the real-time voice service and voice messaging and speech-to-text services, which all depend on core APIs such as `Init` and `Poll`.

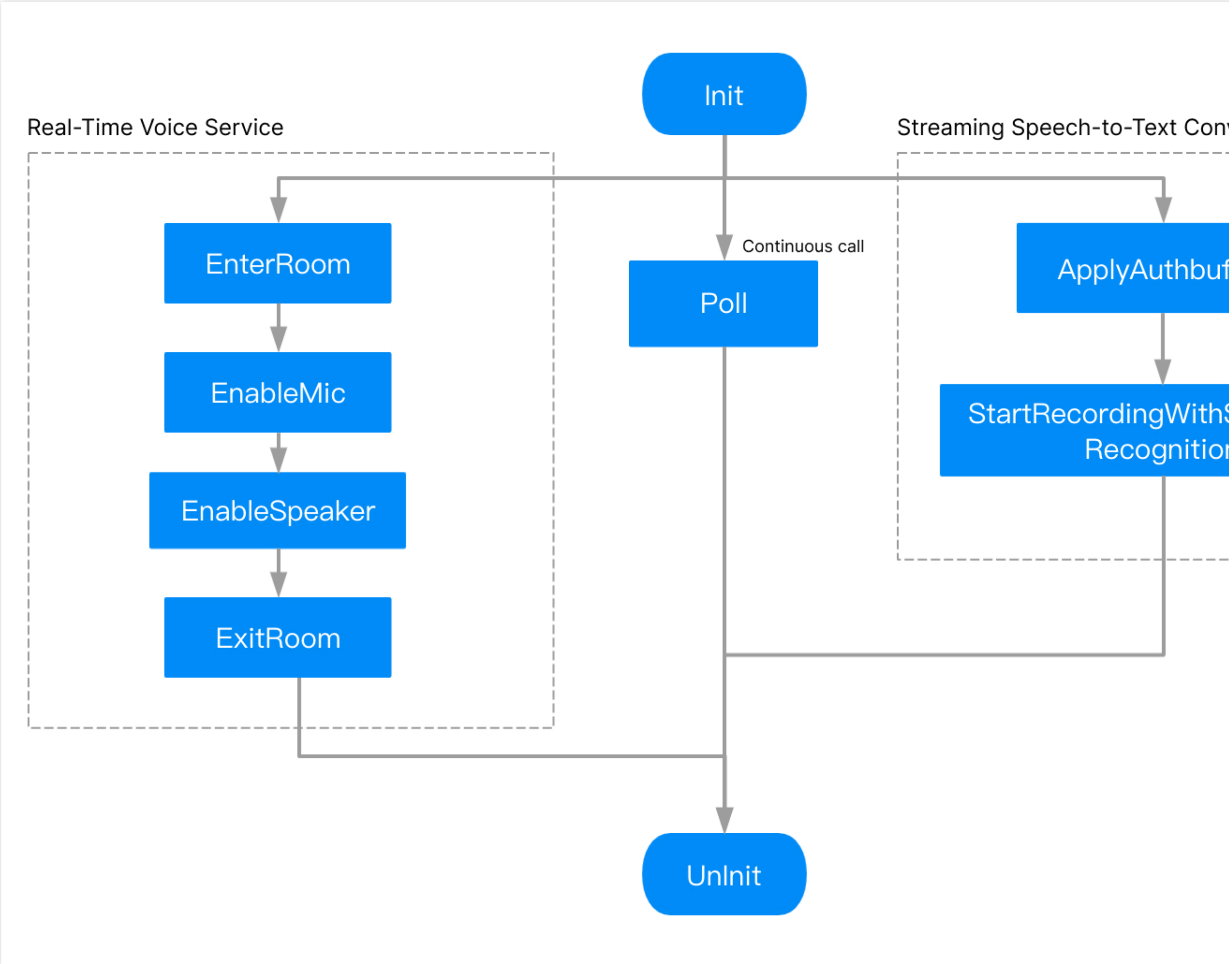
Notes

You have created a GME application and obtained the SDK `AppID` and key. For more information, see Activating Services.
You have activated **GME real-time voice and voice messaging and speech-to-text services**. For more information, see Activating Services.
Configure your project before using GME; otherwise, the SDK will not take effect.
After a GME API is called successfully, `GmeError.AV_OK` will be returned with the value being `0`.
GME APIs should be called in the same thread.
The `Poll` API should be called periodically for GME to trigger event callbacks.
For detailed error codes, see Error Codes.
Note:
There is a default call rate limit for speech-to-text APIs. For more information on how calls are billed within the limit, see Purchase Guide. If you want to increase the limit or learn more about how excessive calls are billed, submit a ticket.
Non-streaming speech-to-text API ***SpeechToText()***: There can be up to 10 concurrent requests per account.
Streaming speech-to-text API ***StartRecordingWithStreamingRecognition()***: There can be up to 50 concurrent requests per account.

Integrating the SDK

Directions

Key processes involved in SDK integration are as follows:



1. Initializing GME
2. Call `Poll` periodically to trigger event callbacks.
3. Initialize authentication.
4. Start streaming speech recognition.
5. Stop recording.
6. Uninitialize GME.

.dart files

```
gme.dart       GME business implementation APIs
gmeType.dart   GME type definition file
gmeError.dart  GME error type definition file
```

Core APIs

API	Description
InitSDK	Initializes GME.
Poll	Triggers the event callback.
Uninit	Uninitializes GME.

Importing the GME module

```
import 'package:gme/gme.dart';
import 'package:gme/gmeType.dart';
```

Getting an instance

var m_context = await ITMGContext.GetInstance();

Initializing the SDK

You need to initialize the SDK through the `Init` API before you can use the real-time voice chat, voice messaging, and speech-to-text services. The `Init` API must be called in the same thread as other APIs. We recommend you call all APIs in the main thread.

API prototype

```
Future<int> InitSDK(String appID, String openID)
```

Parameter	Type	Description
sdkAppId	string	\`AppID\` provided in the GME console, which can be obtained as instructed in Activating Services.
openID	string	\`openID\` can only be in \`Int64\` type, which is passed in after being converted to a string. You can customize its rules, and it must be unique in the application. To pass in \`openID\` as a string, submit a ticket for application.

Returned values

Returned Value	Description
GmeError.AV_OK= 0	SDK initialized successfully.
AV_ERR_SDK_NOT_FULL_UPDATE=7015	Solution: Check whether the SDK file is complete. We recommend that you delete it and then import the SDK again.

Note:

Notes on 7015 error code

The 7015 error code is identified by MD5. If this error is reported during integration, check the integrity and version of the SDK file as prompted.

The returned value `AV_ERR_SDK_NOT_FULL_UPDATE` is **only a reminder** but will not cause an initialization failure.

Sample code

```
int res = await ITMGContext.GetInstance().InitSDK(_editAppID.text,_editOpenID.text)
// Determine whether the initialization is successful by the returned value
if (ret != GmeError.AV_OK)
{
    print("Failed to initialize the SDK:");
    return;
}
```

Triggering event callback

Event callbacks can be triggered by calling the `Poll` API in the timer. The `Poll` API is GME's message pump and should be called periodically for GME to trigger event callbacks; otherwise, the entire SDK service will run abnormally. For more information, see the `EnginePollHelper` file in SDK Download Guide.

Note:

The `Poll` API must be called periodically and in the main thread to avoid abnormal API callbacks.

API prototype

```
Future<void> Poll()
```

Sample code

```
Future<void> pollTimer() async {
  _pollTimer = Timer.periodic(Duration(milliseconds: 100), (Timer timer) {
  ITMGContext.GetInstance().Poll();
  });
}
```

Uninitializing SDK

This API is used to uninitialize the SDK to make it uninitialized. **If the game business account is bound to `openid`, switching game account requires uninitializing GME and then using the new `openid` to initialize again.**

API prototype

```
Future<int> Uninit()
```

Voice Messaging and Speech-to-Text Services

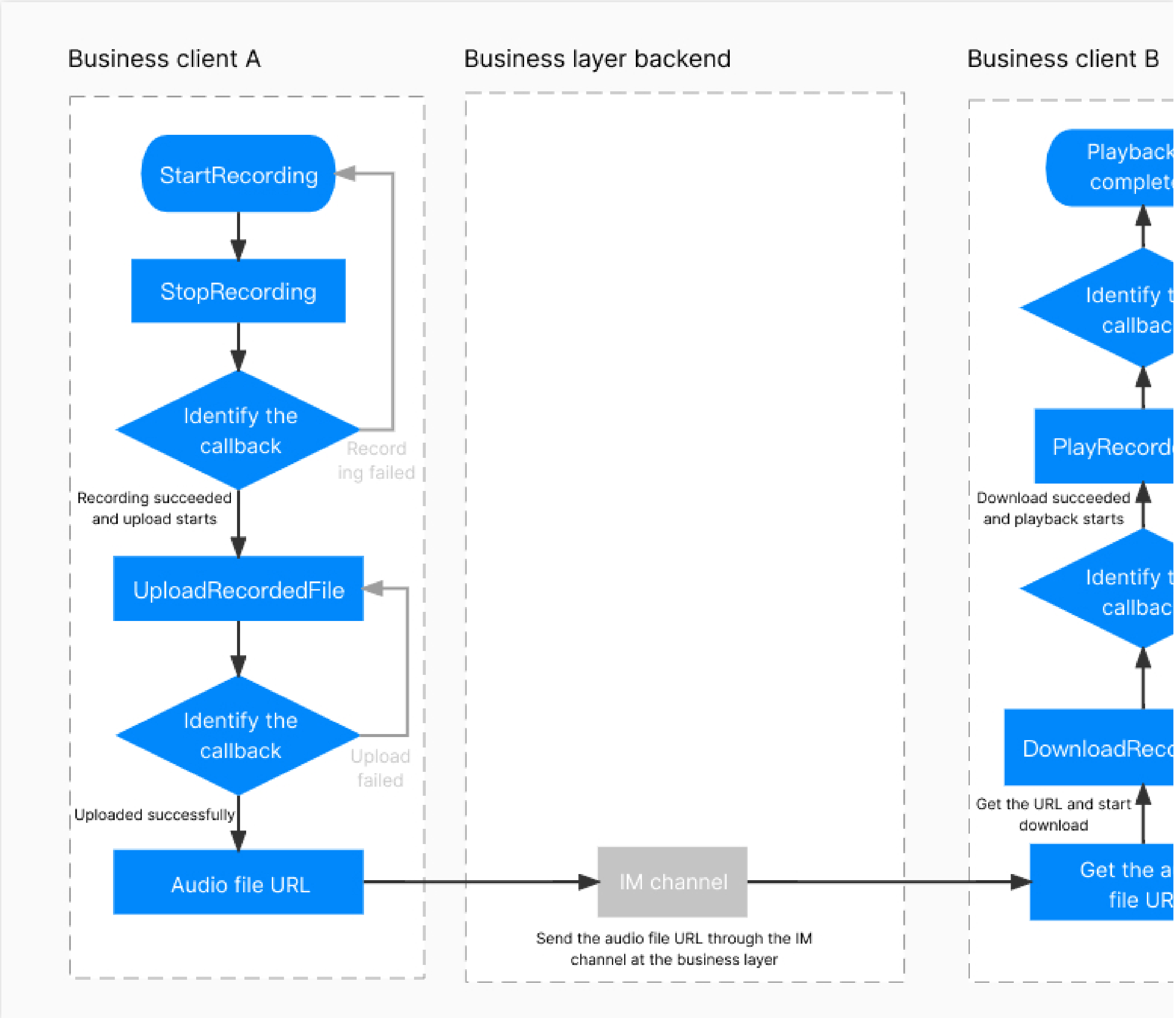
Note:

The speech-to-text service consists of fast recording-to-text conversion and streaming speech-to-text conversion.

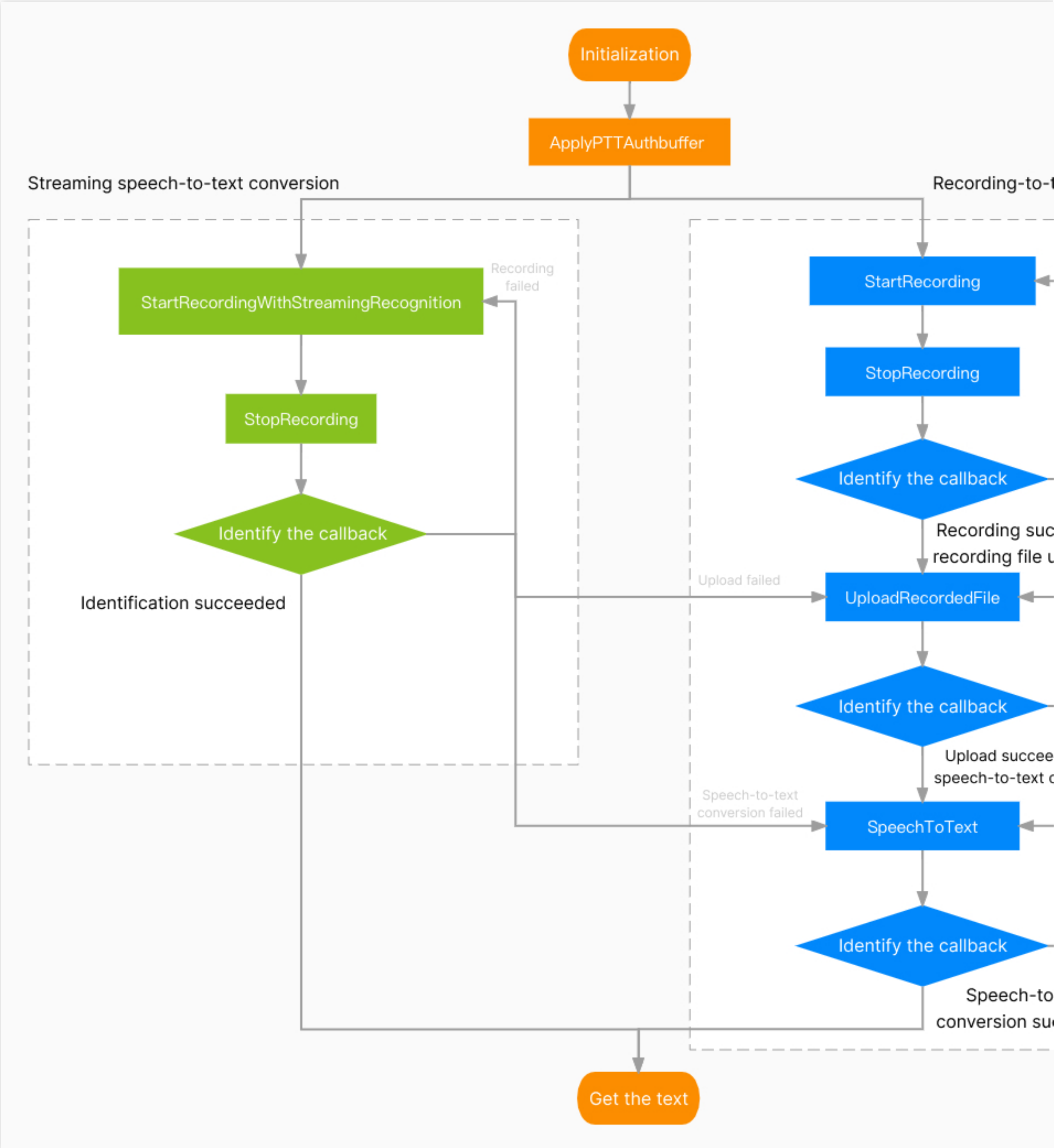
You do not need to enter a voice chat room when using the voice messaging service.

The maximum recording duration of a voice message is 58 seconds by default, and the minimum recording duration cannot be less than 1 second. If you want to customize the recording duration, for example, to change the maximum recording duration to 10 seconds, call the `SetMaxMessageLength` API to set it after initialization.

Flowchart for using the voice message service



Flowchart for using the speech-to-text service



API	Description
GenAuthBuffer	Gets the authentication information.
SetMaxMessageLength	Specifies the maximum duration of a voice message.

Generating the local authentication key

Generate `AuthBuffer` for encryption and authentication of relevant features. For release in the production environment, use the backend deployment key as detailed in Authentication Key.

API prototype

```
Future<Uint8List> GenAuthBuffer(String appID, String roomID, String openID, String
```

Parameter	Type	Description
appId	string	`AppId` from the Tencent Cloud console
roomId	string	Enter `null` or an empty string.
openId	string	User ID, which is the same as `OpenId` during initialization.
key	string	Permission key from the Tencent Cloud console.

Application authentication

After the authentication information is generated, the authentication is assigned to the SDK.

API prototype

```
Future<int> ApplyPTTAuthbuffer(Uint8List authBuffer)
```

Sample code

```
Uint8List authBuffer = await ITMGContext.GetInstance().GenAuthBuffer(_editAppID.tex
m_context.ApplyPTTAuthbuffer(authBuffer);
```

Specifying the maximum duration of voice message

This API is used to specify the maximum duration of a voice message, which can be up to 58 seconds.

API prototype

```
Future<int> SetMaxMessageLength(int msTime)
```

Parameter	Type	Description
msTime	number	Voice message duration in ms. Value range: 1000 < `msTime` <= 58000

Sample code

```
ITMGContext.GetInstance().GetPTT().SetMaxMessageLength(fileLen);
```

Streaming Speech Recognition

Voice messaging and speech-to-text APIs

API	Description
StartRecordingWithStreamingRecognition	Starts streaming recording.
StopRecording	Stops recording.

Starting streaming speech recognition

This API is used to start streaming speech recognition. Text obtained from speech-to-text conversion will be returned in real time in its callback. It can specify a language for recognition or translate the text recognized in speech into a specified language and return the translation. **To stop recording, call** StopRecording**.**

API prototype

```
Future<int> StartRecordingWithStreamingRecognition(String filePath, String speechLa
```

Parameter	Type	Description
filePath	string	Path of the stored audio file
speechLanguage	string	The language in which the voice message file is to be converted to text. For parameters, see Language Parameter Reference List.
translateLanguage	string	The language in which the voice message file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
string filePath = "xx/xxx/xxx.silk"
int res = await ITMGContext.GetInstance().GetPTT().StartRecordingWithStreamingRecog
if (ret == 0) {
    this.currentStatus = "Start streaming recording";
} else {
    this.currentStatus = "Failed to start streaming recording";
}
```

Note:

Translation incurs additional fees. For more information, see Purchase Guide.

Callback for streaming speech recognition

After streaming speech recognition is started, you need to listen on callback messages in the `OnEvent` notification, which is as detailed below:

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COMPLETE` returns text after the recording is stopped and the recognition is completed, which is equivalent to returning the recognized text after a paragraph of speech.

`ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING` returns the recognized text in real time during the recording, which is equivalent to returning the recognized text while speaking.

The event message will be identified in the callback notification based on the actual needs. The passed parameters include the following four messages.

Message	Description
result	Return code indicating whether streaming speech recognition is successful
text	Text converted from speech
file_path	Local path of the stored recording file
file_id	Backend URL address of the recording file, which will be retained for 90 days

Note:

The `file_id` is empty when the 'ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_RUNNING' message is listened for.

Error codes

Error Code	Description	Solution
32775	Streaming speech-to-text conversion failed, but recording succeeded.	Call the `UploadRecordedFile` API to upload the recording file and then call the `SpeechToText` API to perform speech-to-text conversion.
32777	Streaming speech-to-text conversion failed, but recording and upload succeeded.	The message returned contains a backend URL after successful upload. Call the `SpeechToText` API to perform speech-to-text conversion.
32786	Streaming speech-to-text conversion failed.	During streaming recording, wait for the execution result of the streaming recording API to return.
32787	Speech-to-text conversion	Activate the text translation service in the console.

	succeeded, but the text translation service was not activated.	
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

If the error code 4098 is reported, see Speech-to-text Conversion for solutions.

Sample code

```
void handleEventMsg(int eventType, String data){
   switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_COM
        {
            HandleSTREAM2TEXTComplete(data,true);
            break;
        }
        ...
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_STREAMINGRECOGNITION_IS_
        {
            HandleSTREAM2TEXTComplete(data, false);
            break;
        }
    }
}
ITMGContext.GetInstance().SetEvent(handleEventMsg);
```

Voice Message Recording

The recording process is as follows: start recording > stop recording > return recording callback > start the next recording.

Voice messaging and speech-to-text APIs

API	Description
StartRecording	Starts recording.
PauseRecording	Pauses recording.
ResumeRecording	Resumes recording.

StopRecording	Stops recording.
CancelRecording	Cancels recording.

Starting recording

This API is used to start recording.

API prototype

```
Future<int> StartRecording(String filePath)
```

Parameter	Type	Description
filePath	string	Path of the stored voice message file

Sample code

```
string filepath = "xxxx/xxx.silk";
int res = await ITMGContext.GetInstance().GetPTT().StartRecording(filepath);
```

Stopping recording

This API is used to stop recording. It is async, and a callback for recording completion will be returned after recording stops. A recording file will be available only after recording succeeds.

API prototype

```
Future<int> StopRecording()
```

Sample code

```
ITMGContext.GetInstance().GetPTT().StopRecording();
```

Callback for recording start

A callback will be executed through a delegate function to pass a message when recording is completed.

To stop recording, call `StopRecording` . The callback for recording start will be returned after the recording is stopped.

Parameter	Type	Description
code	string	`0`: Recording is completed.

filepath	string	Path of the stored recording file, which must be accessible and cannot be the `fileid`.

Error codes

Error Code	Cause	Solution
4097	A parameter is empty.	Check whether the API parameters in the code are correct.
4098	An initialization error occurred.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
4099	Recording is in progress.	Make sure that the SDK recording feature is used at the right time.
4100	No audio data is captured.	Check whether the mic is working properly.
4101	An error occurred while accessing the file during recording.	Ensure the existence of the file and the validity of the file path.
4102	The mic is not authorized.	The mic permission is required for using the SDK. To add the permission, see the SDK project configuration document for the corresponding engine or platform.
4103	The recording duration is too short.	The recording duration should be in ms and longer than 1,000 ms.
4104	No recording operation is started.	Check whether the recording starting API has been called.

Sample code

```
void handleEventMsg(int eventType, String data){
   switch (eventType) {
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
          // Process
          break;
        }
        ...
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_RECORD_COMPLETE:
```



```
          {
            // Process
            break;
          }
      }
}
ITMGContext.GetInstance().SetEvent(handleEventMsg);
```

Pausing recording

This API is used to pause recording. If you want to resume recording, call the `ResumeRecording` API.

API prototype

```
Future<int> PauseRecording()
```

Sample code

```
ITMGContext.GetInstance().GetPTT().PauseRecording();
```

Resuming recording

This API is used to resume recording.

API prototype

```
Future<int> ResumeRecording()
```

Sample code

```
ITMGContext.GetInstance().GetPTT().ResumeRecording();
```

Canceling recording

This API is used to cancel recording. **There is no callback after cancellation**.

API prototype

```
Future<int> CancelRecording()
```

Sample code

```
ITMGContext.GetInstance().GetPTT().CancelRecording();
```

Voice Message Upload, Download, and Playback

API	Description
UploadRecordedFile	Uploads an audio file.
DownloadRecordedFile	Downloads an audio file.
PlayRecordedFile	Plays back audio.
StopPlayFile	Stops playing back audio.
GetFileSize	Gets the audio file size.
GetVoiceFileDuration	Gets the audio file duration.

Uploading an audio file

This API is used to upload an audio file.

API prototype

```
Future<int> UploadRecordedFile(String filePath)
```

Parameter	Type	Description
filePath	String	Path of the uploaded audio file, which is a local path.

Sample code

```
ITMGContext.GetInstance().GetPTT().UploadRecordedFile(_filePath);
```

Callback for audio file upload completion

After the audio file is uploaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	number	`0`: Recording is completed.

filepath	string	Path of the stored recording file
fileid	string	File URL

Error codes

Error Code	Cause	Solution
8193	An error occurred while accessing the file during upload.	Ensure the existence of the file and the validity of the file path.
8194	Signature verification failed.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
8195	A network error occurred.	Check whether the device can access the internet.
8196	The network failed while getting the upload parameters.	Check whether the authentication is correct and whether the device network can normally access the internet.
8197	The packet returned during the process of getting the upload parameters is empty.	Check whether the authentication is correct and whether the device network can normally access the internet.
8198	Failed to decode the packet returned during the process of getting the upload parameters.	Check whether the authentication is correct and whether the device network can normally access the internet.
8200	No `appinfo` is set.	Check whether the `apply` API is called or whether the input parameters are empty.

Sample code

```
void handleEventMsg(int eventType, String data){
 switch (eventType) {
        ...
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_UPLOAD_COMPLETE:
        {
            // Process
            break;
        }
    }
}
```

```
ITMGContext.GetInstance().SetEvent(handleEventMsg);
```

Downloading the audio file

This API is used to download an audio file.

API prototype

```
Future<int> DownloadRecordedFile(String fileId, String filePath)
```

Parameter	Type	Description
fileId	string	File URL
filePath	string	Local path of the saved file, which must be accessible and cannot be the `fileid`.

Sample code

```
ITMGContext.GetInstance().GetPTT().DownloadRecordedFile(_fileId, _filePath);
```

Callback for audio file download completion

After the audio file is downloaded, the event message `ITMG_MAIN_EVNET_TYPE_PTT_DOWNLOAD_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` , and `file_id` .

Parameter	Type	Description
result	number	`0`: Download is completed.
filepath	string	Path of the stored recording file
fileid	string	URL of the recording file, which will be retained on the server for 90 days.

Error codes

Error Code	Cause	Solution
12289	An error occurred while accessing the file during download.	Check whether the file path is valid.
12290	Signature verification failed.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text

		feature is initialized.
12291	A network storage system exception occurred.	The server failed to get the audio file. Check whether the API parameter `fileid` is correct, whether the network is normal, and whether the file exists in COS.
12292	A server file system error occurred.	Check whether the device can access the internet and whether the file exists on the server.
12293	The HTTP network failed while getting the download parameters.	Check whether the device can access the internet.
12294	The packet returned during the process of getting the download parameters is empty.	Check whether the device can access the internet.
12295	Failed to decode the packet returned during the process of getting the download parameters.	Check whether the device can access the internet.
12297	No `appinfo` is set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.

Sample code

```
void handleEventMsg(int eventType, String data){
    switch (eventType) {
        ...
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
          // Process
          break;
        }
    }
}
ITMGContext.GetInstance().SetEvent(handleEventMsg);
```

Playing back audio

This API is used to play back audio.

API prototype

```
Future<int> PlayRecordedFile(String filePath, int voiceType)
```

Parameter	Type	Description
filePath	string	Local audio file path
voicetype	ITMG_VOICE_TYPE	Voice changing type. For more information, see Voice Changing.

Error codes

Error Code	Cause	Solution
20485	Playback is not started.	Ensure the existence of the file and the validity of the file path.

Sample code

```
int res = await ITMGContext.GetInstance().GetPTT().PlayRecordedFile(_filePath, _nVo
```

Callback for audio playback

After the audio is played back, the event message `ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE` will be returned, which will be identified in the `OnEvent` function.

The passed parameter includes `result` and `file_path` .

Parameter	Type	Description
code	number	\`0\`: Playback is completed.
filepath	string	Path of the stored recording file

Error codes

Error Code	Cause	Solution
20481	An initialization error occurred.	Check whether the device is being used, whether the permissions are normal, and whether the initialization is normal.
20482	During playback, the client tried to interrupt and play back the next one but failed (which should succeed normally).	Check whether the code logic is correct.
20483	A parameter is empty.	Check whether the API parameters in the code are correct.

20484	An internal error occurred.	An error occurred while initializing the player. This error code is generally caused by failure in decoding, and the error should be located with the aid of logs.

Sample code

```
void handleEventMsg(int eventType, String data){
 switch (eventType) {
    case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_PLAY_COMPLETE:
    {
        // Process
        break;
    }
   }
}
```

Stopping audio playback

This API is used to stop audio playback. There will be a callback for playback completion when the playback stops.

API prototype

```
Future<int> StopPlayFile()
```

Sample code

```
ITMGContext.GetInstance().GetPTT().StopPlayFile();
```

Getting audio file size

This API is used to get the size of an audio file.

API prototype

```
Future<int> GetFileSize(String filePath)
```

Parameter	Type	Description
filePath	string	Path of the audio file, which is a local path

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().GetFileSize(_filePath);
```

Getting audio file duration

This API is used to get the duration of an audio file in milliseconds.

API prototype

```
Future<int> GetVoiceFileDuration(String filePath)
```

Parameter	Type	Description
filePath	string	Path of the audio file, which is a local path

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().GetVoiceFileDuration(_file
```

Fast Recording-to-Text Conversion

Translating audio file into text in specified language

This API can specify a language for recognition or translate the text recognized in speech into a specified language and return the translation.

Note：

Translation incurs additional fees. For more information, see Purchase Guide.

API prototype

```
Future<int> SpeechToText(String fileId, String speechLanguage, String translateLang
```

Parameter	Type	Description
fileID	string	URL of the audio file, which will be retained on the server for 90 days.
speechLanguage	string	The language in which the audio file is to be converted to text. For parameters, see Language Parameter Reference List.
translatelanguage	string	The language in which the audio file is to be translated to text. For parameters, see Language Parameter Reference List.

Sample code

```
ITMGContext.GetInstance().GetPTT().SpeechToText(_fileId, "cmn-Hans-CN", "cmn-Hans-C
```

Callback for recognition

After the specified audio file is converted to text, the event message ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE will be returned, which will be identified in the `OnEvent` function.

The passed parameters include `result` , `file_path` and `text` (recognized text).

Parameter	Type	Description
result	number	\`0\`: Recording is completed.
fileid	string	URL of the audio file, which will be retained on the server for 90 days.
text	string	Converted text

Error codes

Error Code	Cause	Solution
32769	An internal error occurred.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32770	A network error occurred.	Check whether the device can access the internet.
32772	Failed to decode the returned packet.	Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32774	No \`appinfo\` is set.	Check whether the authentication key is correct and whether the voice messaging and speech-to-text feature is initialized.
32776	\`authbuffer\` check failed.	Check whether \`authbuffer\` is correct.
32784	The speech-to-text conversion parameter is incorrect.	Check whether the API parameter \`fileid\` in the code is empty.
32785	Speech-to-text translation returned an error.	An error occurred in the voice messaging and speech-to-text feature on the backend. Analyze logs, get the actual error code returned from the backend to the client, and ask backend personnel for assistance.
32787	Speech-to-text conversion	Activate the text translation service in the console.

	succeeded, but the text translation service was not activated.	
32788	Speech-to-text conversion succeeded, but the language parameter of the text translation service was invalid.	Check the parameter passed in.

Sample code

```
void handleEventMsg(int eventType, String data){
 switch (eventType) {
    case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVENT_TYPE_ENTER_ROOM:
        {
          // Process
          break;
        }
        ...
        case ITMG_MAIN_EVENT_TYPE.ITMG_MAIN_EVNET_TYPE_PTT_SPEECH2TEXT_COMPLETE:
        {
          // Process
          break;
        }
    }
}
```

Voice Message Volume Level APIs

API	Description
GetMicLevel	Gets the real-time mic volume level.
SetMicVolume	Sets the recording volume level.
GetMicVolume	Gets the recording volume level.
GetSpeakerLevel	Gets the real-time speaker volume level.
SetSpeakerVolume	Sets the playback volume level.
GetSpeakerVolume	Gets the playback volume level.

Getting the real-time mic volume of voice message

This API is used to get the real-time mic volume. A number-type value will be returned. Value range: 0-200.

API prototype

```
Future<int> GetMicLevel()
```

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().GetMicLevel();
```

Setting the recording volume of voice message

This API is used to set the recording volume of voice message. Value range: 0-200.

API prototype

```
Future<int> SetMicVolume(int volume)
```

Parameter	Type	Description
vol	number	Value range: 0-200. Default value: `100`. `0` indicates that the audio is mute, while `100` indicates that the volume level remains unchanged.

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().SetMicVolume(100);
```

Getting the recording volume of voice message

This API is used to get the recording volume of a voice message. A number-type value will be returned. Value range: 0-200.

API prototype

```
Future<int> GetMicVolume()
```

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().GetMicVolume();
```

Getting the real-time speaker volume of voice message

This API is used to get the real-time speaker volume. A number-type value will be returned. Value range: 0-200.

API prototype

```
Future<int> GetSpeakerLevel()
```

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().GetSpeakerLevel();
```

Setting the playback volume of voice message

This API is used to set the playback volume of voice message. Value range: 0-200.

API prototype

```
Future<int> SetSpeakerVolume(int volume)
```

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().SetSpeakerVolume(100);
```

Getting the playback volume of voice message

This API is used to get the playback volume of a voice message. A number-type value will be returned. Value range: 0-200.

API prototype

```
Future<int> GetSpeakerVolume()
```

Sample code

```
final int res = await ITMGContext.GetInstance().GetPTT().GetSpeakerVolume();
```

Advanced APIs

Getting version number

This API is used to get the SDK version number for analysis.

API prototype

```
Future<String> GetSDKVersion()
```

Sample code

```
_sdkVersions = await ITMGContext.GetInstance().GetSDKVersion();
```

Setting log printing level

This API is used to set the level of logs to be printed, and needs to be called before the initialization. It is recommended to keep the default level.

API prototype

```
Future<int> SetLogLevel(int levelWrite, int levelPrint)
```

Parameter description

Parameter	Type	Description
level	ITMG_LOG_LEVEL	Sets the log level. `TMG_LOG_LEVEL_NONE` indicates not to log. Default value: `TMG_LOG_LEVEL_INFO`.

`level` description:

level	Description
TMG_LOG_LEVEL_NONE	Does not print logs
TMG_LOG_LEVEL_ERROR	Prints error logs (default)
TMG_LOG_LEVEL_INFO	Prints info logs
TMG_LOG_LEVEL_DEBUG	Prints debug logs
TMG_LOG_LEVEL_VERBOSE	Prints verbose logs

Sample code

```
ITMGContext.GetInstance().SetLogLevel(ITMG_LOG_LEVEL.TMG_LOG_LEVEL_ERROR,ITMG_LOG_L
```

Setting the log printing path

This API is used to set the log printing path. The default path is as follows. It needs to be called before Init.

API prototype

```
Future<int> SetLogPath(String logDir)
```

Parameter	Type	Description
logPath	string	Path

Sample code

```
String logDir = ""// Set a path by yourself
ITMGContext.GetInstance().SetLogPath(curPath);
```


SDK Version Upgrade Guide

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document describes the upgrade of GME.

Upgrade from GME 2.x to 2.9

SDK updates

Dynamic library split

Rename Android package

The GME SDK is updated with the following new library files in addition to libgmesdk.

Library files' corresponding features

The new version of GME splits the dynamic libraries to reduce the package size. You can only import the library files you need. For example, if you only need the voice changing feature, to import `libgme_soundtouch` is good.

Library File	Feature
libgmefdkaac	1. Used to enter an SD or HD voice room. 2. Used to play back accompaniment files in ACC format
libgmefaad2	Used to play back accompaniment files in MP4 format
libgmeogg	Used to play back accompaniment files in OGG format
libgmelamemp3	Used to play back accompaniment files in MP3 format
libgmesoundtouch	Used for voice changing and pitch changing

Upgrade Notice

For iOS upgrade, please see iOS Project Upgrade guide.

For Android upgrade, you need to rename package(change Tencent into GME) and modify obfuscation configuration. Please see Project Export.

For Unity upgrade, if you used SD or HD sound quality, or accompaniment, please see Using HD Sound Quality.

Upgrade from GME 2.2 to 2.3.5

SDK updates

New features

Offline voice can be used during voice chat now.

Voice chat can filter offensive, insecure, or inappropriate information.

HTML5-based voice chat is supported now, making voice chat available across all operating systems.

Android v8a architecture is supported now.

Low-latency capture and playback is adaptive to Android now.

Optimizations

Optimized the range voice APIs of the SDK to lower the access threshold.

Optimized noise reduction for voice.

Greatly reduced memory usage by the SDK.

Changes in Major APIs

EnterRoom

The room entering operation has been changed from sync to async. If the return value is 0, the async delivery is successful and waiting to be processed by the callback function; otherwise, the async delivery fails.

```
public abstract int EnterRoom();
```

ExitRoom

The room exiting operation has been changed from sync to async. It is handled in the same way as the RoomExitComplete callback function. If the return value is AV_OK, the async delivery is successful.

Note:

If there is a scenario in the application where room entry is performed immediately after room exit, you don't need to wait for the `RoomExitComplete` callback notification from the `ExitRoom` API during API call; instead, you can directly call the API.

```
public abstract int ExitRoom();
```

Changes in Error Codes

For uniform processing of all error codes, use !AV_OK.

To handle the errors separately, focus on the type of error returned by the API.

Note:

Error code "1" has no specific meaning and will no longer be returned since v2.3.5, so it has been deleted.

Changes in Other APIs

PauseAudio/ResumeAudio


```
public int PauseAudio()
public int ResumeAudio()
```

If the `ITMGAudioCtrl::PauseAudio` or `ResumeAudio` API is called in an SDK before v2.3, please see the table below for version comparison.

Before v2.3	v2.3
For mutual exclusivity with other modules	Change PauseAudio to Pause and change ResumeAudio to Resume
For using offline voice in voice chat	Delete PauseAudio and ResumeAudio

Changes in the Parameters of the SetLogLevel API

Original API

```
ITMGContext virtual void SetLogLevel(int logLevel, bool enableWrite, bool enablePri
```

New API

```
ITMGContext virtual void SetLogLevel(ITMG_LOG_LEVEL levelWrite, ITMG_LOG_LEVEL leve
```

Parameter description

Parameter	Type	Description
levelWrite	ITMG_LOG_LEVEL	Sets the level of logs to be written, TMG_LOG_LEVEL_NONE means not to write
levelPrint	ITMG_LOG_LEVEL	Sets the level of logs to be printed, TMG_LOG_LEVEL_NONE means not to print

ITMG_LOG_LEVEL Type

ITMG_LOG_LEVEL	Description
TMG_LOG_LEVEL_NONE=0	Do not print logs
TMG_LOG_LEVEL_ERROR=1	Prints error logs (default)
TMG_LOG_LEVEL_INFO=2	Prints prompt logs
TMG_LOG_LEVEL_DEBUG=3	Prints development and debugging logs
TMG_LOG_LEVEL_VERBOSE=4	Prints high-frequency logs

Upgrade from GME 2.3.5 to 2.5.1

New APIs

GetSendStreamLevel

This API is used to get the real-time audio upstreaming volume level. An int-type value will be returned. Value range: 0–100.

```
ITMGContextGetInstance()->GetAudioCtrl()->GetSendStreamLevel();
```

GetRecvStreamLevel

This API is used to get the real-time audio downstreaming volume levels of other members in the room. An int-type value will be returned. Value range: 0–100.

```
iter->second.level = ITMGContextGetInstance()->GetAudioCtrl()->GetRecvStreamLevel(i
```

API changes

Type change for returned values of voice messaging and speech-to-text APIs

The type of returned values of the following APIs has been changed to `int` .

```
StartRecording
UploadRecordedFile
DownloadRecordedFile
PlayRecordedFile
SpeechToText
```

Upgrade from GME 2.5 to 2.7

New APIs

PlayRecordedFile(const char* filePath, ITMG_VOICE_TYPE voiceType)

This API is used to playback voice message with voice changing effects.

SetAccompanyKey(int nKey)

This API is used to set the voice chat accompaniment up and down.

에러 코드

최종 업데이트 날짜: : 2024-12-05 15:45:11

본문에서는 개발자가 GME API를 쉽게 디버그하고 액세스할 수 있도록 Tencent Cloud GME(게임 멀티미디어 엔진) 개발 중에 보고될 수 있는 오류 코드를 제공합니다.

일반적인 에러

에러 코드 이름	에러 코드 값	원인 및 제안 솔루션
AV_ERR_3DVOICE_ERR_NOT_INITED	7003	InitSpatializer API를 먼저 호출해야 함
AV_ERR_NET_REQUEST_FALLED	7004	주로 불안정한 네트워크 연결로 인한 네트워크 요청 실패, 문제 해결은 FAQ > Sound and Audio 참고
AV_ERR_CHARGE_OVERDUE	7005	연체로 인한 작업 실패, Tencent Cloud 콘솔에서 계정이 연체되었는지 확인 필요
AV_ERR_AUTH_FIALD	7006	인증 실패, 가능한 원인: 1. 'AppID'가 존재하지 않거나 올바르지 않음 2. 'authbuff' 인증 중 오류 발생 3. 인증 만료
AV_ERR_IN_OTHER_ROOM	7007	이미 다른 방에 있음
AV_ERR_NO_PERMISSION	7009	작업을 수행할 권한이 없음
AV_ERR_FILE_CANNOT_ACCESS	7010	파일에 액세스할 수 없음
AV_ERR_FILE_DAMAGED	7011	파일 손상
AV_ERR_SERVICE_NOT_OPENED	7012	사용하기 전에 콘솔에서 이 기능을 활성화하십시오
AV_ERR_USER_CANCELED	7013	사용자가 방에 들어가는 등의 작업을 성공하기 전에 취소했음
AV_ERR_LOAD_LIB_FAILED	7014	라이브러리 파일이 제대로 로드되지 않았음, 누락되었는지 확인하십시오
AV_ERR_SDK_NOT_FULL_UPDATE	7015	SDK를 업그레이드할 때 모든 파일이 업그레이드되지 않아 일부 모듈 불일치 발생, SDK를 완전히 업그레이드하십시오

AV_ERR_3DVOICE_ERR_FILE_DAMAGED	7002	3D 사운드 파일 로딩에 실패하였습니다

클라이언트 오류

에러 코드 이름	에러 코드 값	의미	원인
AV_ERR_REPEATED_OPERATION	1001	반복 작업	작업이 진행 중일 때 다시 동일한 작업을 실행 시 발생합니다. 작업 유형에는 AVContext, 방, 장치 및 구성원 관련 작업이 포함됩니다. AVContext 유형의 작업: StartContext/StopContext. 방 유형의 작업: EnterRoom/ExitRoom. 장치 유형의 작업: 장치를 시작/종료합니다.
AV_ERR_EXCLUSIVE_OPERATION	1002	독점 작업	동일한 유형의 다른 작업이 진행 중일 때 다시 같은 유형의 작업을 실행 시 해당 에러가 발생됩니다.
AV_ERR_HAS_IN_THE_STATE	1003	반복 작업	대상이 어느 상태에 있을 때 다시 해당 상태로 변경하는 작업 시 해당 에러가 발생됩니다. 예를 들어, 이미 방에 있는 상태에서 다시 방 입장 작업을 실행.
AV_ERR_INVALID_ARGUMENT	1004	잘못된 매개변수	SDK API 호출 시 잘못된 매개변수가 전달되면 해당 에러가 발생됩니다. 예를 들어 방에 입장하려 할 때 전달 방 유형이 AVRoom::ROOM_TYPE_PAIR 또

			는 AVRoom::ROOM_TYPE_MULTI 가 아닐 시 에러가 발생됩니다.
AV_ERR_TIMEOUT	1005	시간 초과	지정된 시간 내에 작업 결과가 반환되지 않을 시 해당 에러가 발생합니다. 이 에러는 주로 시그널 전송 관련된 네트워크 문제가 발생될 때 발생합니다. 예를 들어 방에 들어가기 위한 작업을 수행한 후 30s 이내에 작업 결과가 반환되지 않을 시 에러가 발생합니다.
AV_ERR_NOT_IMPLEMENTED	1006	구현되지 않음	SDK API 호출 시 해당 기능이 지원되지 않는다면 해당 기능을 사용할 수 없습니다.

AV_ERR_NOT_IN_MAIN_THREAD	1007	메인 스레드에 없음	SDK 외부 API는 메인 스레드에서 호출해야 합니다. 만약 메인 스레드에서 호출하지 않으면 해당 에러가 발생합니다.
AV_ERR_RESOURCE_IS_OCCUPIED	1008	리소스 점유됨	필요한 카메라 또는 화면과 같은 필수 리소스가 이미 사용 중 일 때 해당 에러가 발생합니다.
AV_ERR_CONTEXT_NOT_EXIST	1101	AVContext 상태 문제	AVContext 객체가 CONTEXT_STATE_STARTED 상태가 아닐 때 AVContext 객체가 이 상태에 있을 때만 호출할 수 있는 API를 호출하면 에러가 발생합니다.

AV_ERR_CONTEXT_NOT_STOPPED	1102	AVContext 상태 문제	AVContext 객체가 CONTEXT_STATE_STOPPED 상태가 아닐 때 해당 상태에 있을 때만 호출할 수 있는 API를 호출하면 에러가 발생합니다. AVContext::DestroyContext와 같이 AVContext 객체가 해당 상태일 때만 호출할 수 있는 API를 호출하면 에러가 발생합니다.
AV_ERR_ROOM_NOT_EXIST	1201	AVRoom 상태 문제	AVRoom 객체가 ROOM_STATE_ENTERED 상태가 아닐 때 AVRoom 객체가 이 상태일 때만 호출할 수 있는 API를 호출하면 에러가 발생합니다.
AV_ERR_ROOM_NOT_EXITED	1202	AVRoom 상태 문제	AVRoom 객체가 ROOM_STATE_EXITED 상태가 아닐 때 해당 상태에 있을 때만 호출할 수 있는 API를 호출하면 에러가 발생합니다. AVContext::StopContext와 같이 AVRoom 객체가 해당 상태일 때만 호출할 수 있는 API를 호출하면 에러가 발생합니다.
AV_ERR_DEVICE_NOT_EXIST	1301	장치가 존재하지 않음	장치가 존재하지 않거나 초기화되지 않은 장치를 사용 시 에러가 발생합니다.

AV_ERR_ENDPOINT_NOT_EXIST	1401	AVEndpoint 객체가 존재하지 않음	구성원이 음성 채팅 또는 영상 채팅을 시작하지 않은 상태에서 구성원의 AVEndpoint 객체를 얻으려고 시도하였습니다.
AV_ERR_ENDPOINT_HAS_NOT_VIDEO	1402	구성원이 영상 채팅을 시작하지 않음	구성원이 영상 채팅을 시작하지 않은 상태에서 영상이 시작된 후에만 완료할 수 있는 작업을 실행했습니다. 예를 들어, 구성원이 화상 채팅을 시작하지 않은 상태에서 구성원의 화면을 요청했습니다.
AV_ERR_TINYID_TO_OPENID_FAILED	1501	tinyid를 identifier로 변환 실패	구성원의 정보 업데이트를 나타내는 신호를 수신한 후 tinyid를 identifier로 변환해야 합니다. 그러나 IMSDK 라이브러리 또는 네트워크의 로직 문제로 인해 변환에 실패하였습니다.

AV_ERR_OPENID_TO_TINYID_FAILED	1502	identifier를 tinyid로 변환 실패	StartContext API를 호출할 때 클라이언트는 identifier를 tinyid로 변환해야 합니다. 그러나 IMSDK 라이브러리의 로직 문제, 네트워크 문제 또는 미로그인 상태로 인해 변환이 실패하였습니다.
AV_ERR_DEVICE_TEST_NOT_EXIST	1601	AVDeviceTest 상태 문제	AVDeviceTest 객체가 DEVICE_TEST_STATE_STARTED 상태가 아닐 때 AVDeviceTest 객체가 이 상태일 때만 호출할 수 있는 API를 호출했습니다.
AV_ERR_DEVICE_TEST_NOT_STOPPED	1602	AVDeviceTest 상태 문제	AVDeviceTest 객체가 DEVICE_TEST_STATE_STOPPED 상태가 아닐 때 AVDeviceTest 객체가 이 상태일 때만 호출할 수 있는 API를 호출했습니다.(예: API AVContext::StopContext).

AV_ERR_INVITET_FAILED	1801	초대장 발송 실패	초대를 보내려고 하면 실패가 발생합니다.
AV_ERR_ACCEPTT_FAILED	1802	초대 수락 실패	클라이언트가 초대를 수락하려고 하면 실패가 발생합니다.

AV_ERR_REFUSE_FAILED	1803	초대 거부 실패	클라이언트가 초대를 거부하려고 하면 실패가 발생합니다.
QAV_ERR_NOT_TRY_NEW_ROOM	2001	새 방에 들어가지 못하고 원래 방에 남아 있습니다.	새 방으로 전환하는 데 실패하고 현재 방에 남아 있음
QAV_ERR_TRY_NEW_ROOM_FAILED	2002	새 방에 들어가려고 시도하지만 실패하고 원래 방을 나갑니다.	새 방으로 전환하는 데 실패하고 원래 방을 나감

서버 오류

에러 코드 이름	에러 코드	설명	원인	제안된 솔루션

	값			
AV_ERR_SERVER_FAILED	10001	일반적인 오류	백엔드에서 클라이언트로 반환된 실제 오류 코드(로그 내)를 기반으로 특정 원인을 찾습니다.	AppID, UIN 및 AuthBuffer와 같은 방 입장을 위한 API 매개변수의 유효성을 확인합니다(문서 참고). 콘솔에서 관련 매개변수가 로컬 매개변수와 일치하는지 확인하고, 연체된 계정이 있는지 확인합니다. 개발자의 테스트 장치가 내부 네트워크 또는 외부 네트워크에서 실행되고 있는지 확인하십시오.
AV_ERR_SERVER_INVALID_ARGUMENT	10002	잘못된 매개변수	SDK API가 호출되거나 SDK 내부 신호가 백엔드 전송될 때 하나 이상의 잘못된 매개변수가 전달됐습니다.	SDK API 호출에서 전달된 매개변수의 정확성을 확인하십시오. 로그를 분석하고 백엔드에서 클라이언트로 반환된 실제 에러 코드를 얻은 다음 백엔드 담당자에게 도움을 요청하십시오.

AV_ERR_SERVER_NO_PERMISSION	10003	권한 없음	기능을 사용할 권한이 없습니다. 예를 들어, 방에 들어가려고 할 때 올바르지 않거나 만료된 서명을 제공했습니다.	올바른 권한 매개변수를 제공한 후에만 기능을 사용하십시오. AppID와 권한 키가 맞는지 확인하십시오.
AV_ERR_SERVER_TIMEOUT	10004	시간 초과	지정된 시간 내에 작업 결과가 반환되지 않았습니다.	로그를 분석하고 백엔드에서 클라이언트로 반환된 실제 오류 코드를 얻은 다음 백엔드 담당자에게 도움을 요청하십시오.
AV_ERR_SERVER_ALLOC_RESOURCE_FAILED	10005	네트워크 오류	클라이언트가 작업을 수행할 때 네트워크 오	AppID, UIN 및 AuthBuffer와 같은 방 입장을 위한 API 매개변수의 유효성을 확인합니다(문서 참고). 유효한 경우 개발자의 테스트 장치가 내부 네트워크 또는 외부 네트워크에서 실행되고 있는지 확인하십시오. 내부 네트워크인 경우 개발자에게 url: openmsf.3g.qq.com:15000에

			류 발생	연결할 수 있는지 확인하도록 요청하십시오. 만약 그렇다면 url: cloud.tim.qq.com:15000에 연결할 수 있는지 확인하십시오.
AV_ERR_SERVER_ID_NOT_IN_ROOM	10006	방에 없음	클라이언트는 방에 없을 때 일부 작업을 수행했습니다.	SDK 기능이 적시에 사용되는지 확인
AV_ERR_SERVER_NOT_IMPLEMENT	10007	구현되지 않음	SDK API 호출 시 해당 기능을 사용할 수 없습니다.	다른 대체 방법을 찾으십시오.
AV_ERR_SERVER_REPEATED_OPERATION	10008	반복 작업	동일한 유형의 다른 작업이 진행 중일 때 작업을 실행했습니다.	마지막 작업이 완료된 후 작업을 실행합니다.
AV_ERR_SERVER_ROOM_NOT_EXIST	10009	방이	존재하지	SDK 기능이 적시에 사용되는지 확인

		존재하지 않음	않는 방에서 작업을 실행했습니다.	
AV_ERR_SERVER_ENDPOINT_NOT_EXIST	10010	구성원이 존재하지 않음	존재하지 않는 구성원과 관련된 일부 작업이 실행됐습니다.	로그를 분석하고 백엔드에서 클라이언트로 반환된 실제 오류 코드를 얻은 다음 백엔드 담당자에게 도움을 요청하십시오.
AV_ERR_SERVER_INVALID_ABILITY	10011	잘못된 기능	백엔드에서 클라이언트로 반환된 실제 오류 코드(로그 내)를 기반으로 특정 원인을 찾습니다.	로그를 분석하고 백엔드에서 클라이언트로 반환된 실제 오류 코드를 얻은 다음 백엔드 담당자에게 도움을 요청하십시오.

음성 메시지 오류

에러 코드 이름	에러 코드 값	설명	원인
QAVPTTERROR_RECORDER _PARAM_NULL	4097	녹음 오류	매개 가 비 있음
QAVPTTERROR_RECORDER _INIT_ERROR	4098	녹음 오류	초기 류
QAVPTTERROR_RECORDER _RECORDING_ERR	4099	녹음 오류	녹음 중
QAVPTTERROR_RECORDER _NODATA_ERR	4100	녹음 오류	오디 이터 되지
QAVPTTERROR_RECORDER _OPENFILE_ERR	4101	녹음 오류	녹화 액세 오류
QAVPTTERROR_RECORDER _PERMISSION_MIC_ERR	4102	녹음 오류	마이 인증 않았
QAVPTTERROR_RECORDER _AUDIO_TOO_SHORT	4103	녹음 오류	녹음 이 너 짧음
QAVPTTERROR_RECORDER _RECORD_NOT_START	4104	녹음 오류	녹음 이 시 지 않
QAVPTTERROR_UPLOAD _FILE_ACCESSERROR	8193	업로드 오류	파일 스 중 류 발

QAVPTTERROR_UPLOAD _SIGN_CHECK_FAIL	8194	업로드 오류	서명 실패
QAVPTTERROR_UPLOAD _COS_INTERNAL_FAIL	8195	업로드 오류	네트 오류
QAVPTTERROR_UPLOAD _GET_TOKEN_NETWORK_FAIL	8196	업로드 오류	업로 개변 가져 동안 워크 류 발
QAVPTTERROR_UPLOAD _SYSTEM_INNER_ERROR	8197	업로드 오류	업로 개변 가져 동안 된 패 비어
QAVPTTERROR_UPLOAD _RSP_DATA_DECODE_FAIL	8198	업로드 오류	업로 개변 가져 동안 된 패 디코 패
QAVPTTERROR_UPLOAD _APPINFO_UNSET	8200	업로드 오류	appir 설정 않았
QAVPTTERROR_DOWNLOAD _FILE_ACCESSERROR	12289	다운로드 오류	파일 스 중 류 발
QAVPTTERROR_DOWNLOAD _SIGN_CHECK_FAIL	12290	다운로드 오류	서명 실패

QAVPTTERROR_DOWNLOAD _COS_INTERNAL_FAIL	12291	다운로드 오류	네트 오류
QAVPTTERROR_DOWNLOAD _REMOTEFILE_ACCESSERROR	12292	다운로드 오류	서버 시스 류
QAVPTTERROR_DOWNLOAD _GET_SIGN_NETWORK_FAIL	12293	다운로드 오류	다운 매개 를 가 는 동 HTTI 트워 패
QAVPTTERROR_DOWNLOAD _SYSTEM_INNER_ERROR	12294	다운로드 오류	다운 매개 를 가 는 동 반환 킷이 있음
QAVPTTERROR_DOWNLOAD_GET _SIGN_RSP_DATA_DECODE_FAIL	12295	다운로드 오류	다운 매개 를 가 는 동 반환 킷 디 실패
QAVPTTERROR_DOWNLOAD _APPINFO_UNSET	12297	다운로드 오류	appir 설정 않았
QAVPTTERROR_PLAYER_INIT_ERR	20481	재생 오류	초기 류
QAVPTTERROR_PLAYER _PLAYING_ERR	20482	재생 오류	재생 중단

			다음 생하 했지 패했 다(정 으로 해야
QAVPTTERROR_PLAYER _PARAM_NULL	20483	재생 오류	매개 가 비 있음
QAVPTTERROR_PLAYER _OPENFILE_ERR	20484	재생 오류	파일 스 중 생 오 발생
QAVPTTERROR_PLAYER _PLAYER_NOT_START_ERR	20485	재생 오류	재생 작되 음
QAVPTTERROR_S2T _INTERNAL_ERROR	32769	음성-텍스트 변환 오류	내부 발생
QAVPTTERROR_S2T _NETWORK_FAIL	32770	음성-텍스트 변환 오류	네트 실패
QAVPTTERROR_S2T _RSP_DATA_DECODE_FAIL	32772	음성-텍스트 변환 오류	응답 파싱
QAVPTTERROR_S2T _APPINFO_UNSET	32774	음성-텍스트 변환	appi 설정

		오류	않음
QAVPTTERROR_STREAMIN _RECORD_SUC_REC_FAIL	32775	스트리밍 음성 텍스트 변환 오류	스트 음성 트 변 실패 만 녹 성공
QAVPTTERROR_S2T _SIGN_CHECK_FAIL	32776	authbuffer 인증 실패	auth 인증
QAVPTTERROR_STREAMIN _UPLOADANDRECORD_SUC_REC_FAIL	32777	스트리밍 음성 텍스트 변환 오류	스트 음성 트 변 실패 만 녹 및 업 는 성 습니
QAVPTTERROR_S2T_PARAM_NULL	32784	음성-텍스트 변환 오류	잘못 성-텍 변환 변수
QAVPTTERROR_S2T _AUTO_SPEECH_REC_ERROR	32785	음성-텍스트 변환 오류	음성 트 변 서 오 반환
QAVPTTERROR_ERR _VOICE_STREAMIN_RUNING_ERROR	32786	스트리밍 음성-텍스트 변환 오류	스트 음성 트 변 실패
QAVPTTERROR_ERR_VOICE _STREAMING_ASR_ERROR	50012	스트리밍 음성-텍스	요청 오류

		트 변환 오류	

Toolchain

최종 업데이트 날짜: : 2024-01-18 15:15:48

This document lists the toolchain for GME integration.

V2.9.1

Platform	Description
Win32	Windows Software Development Kit (Visual Studio 2015): 10.0.14393.0 Platform Toolset: v140_xp
macOS	MacOS Minimum Deployment Target: 10.10 macOS SDK: 12.0 Xcode®: 13.1 Target Architectures: x86_64
iOS	iOS Minimum Deployment Target: 9.0 iOS SDK: 15.0 Xcode®: 13.1 Target Architectures: arm64, armv7, arm64(simulator), x86_64(simulator)
Android	Android SDK: Minimum API 16 Android SDK Tools: 30.0.3 Android NDK: r23
PS4	SDK: SDK 6.000
PS5	SDK: SDK 3.000
Switch	SDK: DevEnv8.3.0
XboxOne	Microsoft® XDK: 10.0.17134.5055 (July 2018 qfe7)