

Tencent Real-Time Communication クイックスタート 製品ドキュメント



©2013-2019 Tencent Cloud. All rights reserved.



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

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

Trademark Notice

🔗 Tencent Cloud

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

Service Statement

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

カタログ:

クイックスタート SDK クイックインテグレーション iOS Android Mac Windows Web Electron クイックインテグレーション(Flutter) クイックインテグレーション(QT) クイックインテグレーション(Unity) クイックインテグレーション(React Native) クイックインテグレーション(Unreal Engine) Demo クイックスタート Demoクイックスタート(iOS&Mac) Android Demoクイックスタート(Windows) Web Electron 跑通Demo(Flutter) Demoクイックスタート(Unity) Demoクイックスタート(React Native) **Unreal Engine** 入門者のよくあるご質問

クイックスタート SDK クイックインテグレーション iOS

最終更新日:::2022-04-02 16:27:02

このドキュメントでは、主にTRTC SDK(iOS)を迅速にプロジェクトに統合する方法を紹介します。以下のステ ップにしたがって設定するだけで、SDK統合のタスクが完了します。

開発環境要件

- Xcode 9.0+。
- iOS 9.0以上のiPhoneまたはiPadの実機。
- プロジェクトに有効な開発者の署名が設定してあること。

TRTC SDKの統合

CocoaPodsを使用して自動でローディングする方式か、または先にSDKをダウンロードして、それを現在のプロ グラムのプロジェクトにインポートする方式を選択できます。

CocoaPods

1. CocoaPodsのインストール

端末のウィンドウに次のコマンドを入力します(事前にMacにRuby環境をインストールしておく必要があります)。

sudo gem install cocoapods

2. Podfileファイルの作成

プロジェクトが存在するパスに入り、次のコマンドラインを入力するとプロジェクトパスの下にPodfileファイル が現れます。

pod init

3. Podfileファイルの編集



platform :ios, '8.0'

```
target 'App' do
pod 'TXLiteAVSDK_TRTC', :podspec => 'http://pod-1252463788.cosgz.myqcloud.com/liteavsdkspec/TXLit
eAVSDK_TRTC.podspec'
end
```

CocoaPodの公式ソースを使用することもできますが、ダウンロードスピードは遅くなります。

```
platform :ios, '8.0'
source 'https://github.com/CocoaPods/Specs.git'
```

```
target 'App' do
pod 'TXLiteAVSDK_TRTC'
end
```

4. SDKの更新およびインストール

端末のウィンドウに以下のコマンドを入力してローカルライブラリのファイルを更新し、TRTC SDKをインストー ルします。

pod install

または次のコマンドを使用してローカルライブラリのバージョンを更新します。

pod update

podコマンドの実行が完了すると、SDKが統合された.xcworkspaceという拡張子のプログラムファイルが生成されますので、これをダブルクリックして開きます。

説明:

必要なシステムの依存ライブラリAccelerate.frameworkを手動で追加する必要があります。

手動による統合

- 1. TRTC SDKをダウンロードし、ダウンロードが完了したら解凍を行います。
- お客様のXcodeのプロジェクトを開き、動作させたいtargetを選択して、Build Phasesの項目を選択します。

	🔡 < 🗦 📓 TRTCDemo						
TRTCDemo	General	Signing & Capabilities	Resource Tags	Info	Build Settings	Build Rules	
TRTCDemo TRTCDemoTests	PROJECT	+					
Products	TRTCDemo	Dependencies (0 items	0				
▶ Pods	TARGETS						
🔻 🔁 Frameworks		[CP] Check Pods Manife	est.lock				
	TRTCDemoTests	Compile Sources (3 iter	ms)				
		Link Binary With Librari	ies (0 items)				
		Copy Bundle Resources	s (3 items)				

3. Link Binary with Librariesの項目をクリックして展開し、一番下の「+」記号のアイコンをクリックして 依存ライブラリを追加します。

▼ B TRTCDemo	[General	Signing & Capabiliti	es Resource Tags	Info	Build Settings		Build Rules	
TRTCDemo TRTCDemoTests	PROJECT	+						
	TRTCDemo	> Benerative (0 (to mo)					
▶ <mark></mark> Pods	TARGETS	Dependencies (u items)					
🔻 🚬 Frameworks	🕂 TRTCDemo	[CP] Check Pods	s Manifest.lock					
	TRTCDemoTests							
		Compile Source	s (3 items)					
		▼ Link Binary With	Libraries (0 items)					
			Name				Status	
				Ado	d frameworks & librari	ies here		
			+ -		ag to reorder linked b			
		► Copy Bundle Re	sources (3 items)					

4. 順番に、ダウンロードしたTRTC SDK Frameworkおよびそれに必要な依存ライブラリlibc++.tbd、
 Accelerate.frameworkおよびlibresolv.tbd、AVFoundation.frameworkを追加します。



	🔡 < 🗦 🗎 TRTCDen	10							(H	
TRTCDemo	General	Signing & Capabi	ilities Resc	urce Tags	Info	Build Settings		Build Rules		
	PROJECT	+								
	TRTCDemo									
▶ Pods	TARGETS	Dependencie	es (0 items)							
▼ Trameworks	🐴 TRTCDemo	► [CP] Check P	ods Manifest lo	-k						
libc++.tbd	TRTCDemoTests	F [or] oneck F	ous munitestilo	- R						
Accelerate.framework		Compile Source	rces (3 items)							
TXLiteAVSDK_TRTC.framework										
		Link Binary W	▼ Link Binary With Libraries (3 items)							
			Name					Status		
			libc++.tbo	ł				Required 🗘		
			🚘 Accelerat	e.framework				Required 🗘		
			🚔 TXLiteAV	SDK_TRTC.frai	mework			Required 🗘		
			+ -			rag to reorder linked	binaries			
		Copy Bundle	Resources (3 ite	ems)					×	

5. TRTC SDK 9.5.11234以上のバージョンダイナミックライブラリの依存を追加する必要がある。

Generalをクリックし、Frameworks,Libraries,and Embedded Contentを選択します。一番下の「+」アイコンをクリックして順にTXLiteAVSDK_TRTC.frameworkに必要なダイナミックライブラリ
 BoringSSL.xcframework、FFmpeg.xcframework、SoundTouch.xcframeworkを追加し、
 Embed & Signを選択します。

		RTCDemo.xcodeproj					₹ (
V 🖪 TRTCDemo	🛃 TRTCDemo						
> TRTCDemo		General Signing &	Capabilities Resour	rce Tags Info	Build Settings	Build Phases Build Rule	es
> 🚍 Products	PROJECT		App Icons Source	Applcon		0	
> 📷 Frameworks					icon assots		
					o icon assets		
	TARGETS		Launch Screen File	LaunchScreen			
		∽ Supp	orted Intents				
	TRICDemo			Authontication			
			Class Name	Authentication			
				Add intents el	igible for in-app h	andling here	
		∽ Fram	eworks, Libraries, and	d Embedded Cont	ent		
			Name			Embed	
			音 Accelerate.framew	vork		Do Not Embed 🔇	;
			🚘 AVFoundation.fram	nework		Do Not Embed 🔇	;
			🚘 BoringSSL.xcfram	ework		Embed & Sign 🗘	
			🚔 FFmpeg.xcframew	vork		Embed & Sign 🗘	
			■ libc++.tbd				
		6	libresolv.tbd				
		l	SoundTouch.xcfra	mework		Embed & Sign 🗘	
				C.Iramework			<u> </u>
		∽ Deve	lopment Assets				

カメラとマイクの使用権限の許可

SDKの音声ビデオ機能を使用するには、マイクとカメラの使用権限を許可する必要がありますので、Appの Info.plistの中に次の2項目を追加します。システムが使用許可のダイヤログボックスをポップアップするときに 表示されるマイクとカメラ情報にそれぞれ対応します。

- Privacy Microphone Usage Description、さらにマイク使用目的のプロンプトを記入します。
- Privacy Camera Usage Description、さらにカメラ使用目的のプロンプトを記入します。

Demo 🔪 🎢 iPhone 8	TRTCDemo Build TRT	CDemo: Succeede	d Today at 3:28 PM							Ø	\leftrightarrow
踞 < > 🖹 TRTCDemo											
	General	Capabilities	Resource Tags			Build	Settings	Build Phases	Build Rules		
PROJECT	Custom iOS Target	Properties									
A TRTCDemo		Properties									
TARGETS		Кеу			Туре		Value				
		Required device of the second seco	apabilities	0	Array		(1 item)				
introbeline		Privacy - Microph	one Usage Descript	\$							
		Privacy - Camera	Usage Descrip 🔶 🕻	0							
		Bundle identifier		\$	String		\$(PRODUC1	LBUNDLE_IDENTI	FIER)		
		InfoDictionary ver	rsion	٢			6.0				
		Main storyboard f	ile base name	\$			Main				
		Bundle version		\$			1				
		Required backgro	und modes	0			(1 item)				
		Executable file		0			\$(EXECUTA	BLE_NAME)			
		Application requir	es iPhone environm	\$			YES				٢
		Launch screen int	terface file base name	0			LaunchScre	en			
		Bundle display na	me	Ô			腾讯视频通讯	舌			
		Supported interfa	ce orientations	0			(1 item)				
		Bundle versions s	tring, short	Ô			2.0.0				
		Bundle OS Type o	ode	Ô			APPL				
		Localization nativ	e development region	0			\$(DEVELOP	MENT_LANGUAGE	=)		٢
		Supported interfa	ce orientations (iPad)	0			(4 items)				
		Bundle name		\$	String		\$(PRODUCT	I_NAME)			

TRTC SDKの引用

TRTC SDKでは2種類の呼び出し方式をサポートしていますので、いずれかをお選びください

方式1: Objective-CまたはSwiftインターフェースによるTRTC SDKの引用

Objective-CまたはSwiftコードの中でSDKを使用する方式は2種類あります。

• **モジュールの引用**:プロジェクトのSDK APIを使用したいファイルの中に、モジュールを追加して引用しま す。

@import TXLiteAVSDK_TRTC;

• **ヘッダーファイルの引用**:プロジェクトのSDK APIを使用したいファイルの中に、具体的なヘッダーファイル をインポートします。

#import TXLiteAVSDK_TRTC/TRTCCloud.h

方式2:C++インターフェースによるTRTC SDKの引用

ヘッダーファイルの引用: C++インターフェースを使用してiOSアプリケーションを開発したい場合
 は、 TXLiteAVSDK_TRTC. framework/Headers/cpp_interface ディレクトリの下のヘッダーファイルをインポートしてください。

#include TXLiteAVSDK_TRTC/cpp_interface/ITRTCCloud.h

 ネームスペースの利用: C++ののインターフェースのメソッド、タイプなどはいずれもtrtcネームスペースの 中に定義されています。コードをより簡潔にするため、trtc全プラットフォームネームスペースを直接使用す ることをお勧めします

using namespace trtc;

説明:

C++インターフェースの使用方法については、全プラットフォーム(C++)APIの概要をご参照ください。

よくあるご質問

TRTC SDKはバックグラウンドでの動作をサポートしていますか。

サポートしています。バックグラウンドに入っても、依然として関連機能が動作するようにしたい場合は、現在のプログラムのプロジェクトを選択し、**Capabilities**の下の**Background Modes**の設定を**ON**にして、

Audio、AirPlay and Picture in Pictureにチェックを入れます。以下の通りです。



Android

最終更新日:::2022-03-03 14:40:33

このドキュメントでは、主にTRTC SDK(Android)をプロジェクトに迅速に統合する方法を紹介します。以下のス テップにしたがって設定すれば、SDK統合のタスクを完了できます。

開発環境要件

- Android Studio 3.5+。
- Android 4.1 (SDK API 16) およびそれ以上のシステム。

SDKの統合 (aar)

Gradleを使って自動でローディングするか、または手動でaarをダウンロードし、それをプロジェクトにインポートする方式を選択できます。

方法1:自動ローディング(aar)

TRTC SDK はmavenCentralでライブラリを公開していますので、Gradleを設定すれば、自動でダウンロードされ、更新されます。

Android Studioを使って SDKを統合したいプログラム(ここでの例は、TRTCScenesDemo)を開き、その後簡

単な3つのステップで app/build.gradle ファイルを修正しさえすれば、SDKの統合が完成します。



- 1. dependenciesの中にTRTCSDKの依存を追加します。
- 3.xバージョンの com.android.tools.build:gradle のツールを使用している場合は、次のコマンドを実行して ください。

```
dependencies {
implementation 'com.tencent.liteav:LiteAVSDK_TRTC:latest.release'
}
```

 2.xバージョンの com.android.tools.build:gradle のツールを使用している場合は、次のコマンドを実行して ください。

```
dependencies {
  compile 'com.tencent.liteav:LiteAVSDK_TRTC:latest.release'
}
```

2. defaultConfigの中で、Appが使用するCPUアーキテクチャを指定します。

```
defaultConfig {
  ndk {
   abiFilters "armeabi", "armeabi-v7a", "arm64-v8a"
  }
}
```

説明:

現在 TRTC SDKは、armeabi、armeabi-v7a、arm64-v8aをサポートしています。

3. 【Sync Now】をクリックし、自動でSDKをダウンロードし、プロジェクトに統合します。

方法2:手動ダウンロード(aar)

お客様のネットワークとmavenCentralの接続に問題がある場合は、SDKを手動でダウンロードし、プログラム に統合することができます。

- 1. 最新バージョンのTRTC SDKをダウンロードします。
- 2. ダウンロードした aar ファイルをプログラムの**app/libs** ディレクトリの下にコピーします。
- 3. プログラムのルートディレクトリの下の build.gradle の中に、**flatDir**を追加し、ローカルライブラリのパス を指定します。

	TRTCScenesDemo [~/github/TRTCSDK/Android/TRTCScenesDemo] - build.gradle (TRTCScenesDemo)	
Note: TRTCScenesDemo > R build.gradle	▲ app マ No Devices マ ▶ (1) 三 美 単 ⑤ の 裁 目 Git: ビ ✓ ⑥ ち	🖿 🗈 🏘 🖳 🔍 🖂
b Project → 💮 🛧 🍁 -	🔊 build.gradle (TRTCScenesDemo) ×	R
🖉 🔻 🍋 TRTCScenesDemo ~/github/TRTCSDK/Andr	Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.	Sync Now
S gradle	14 b // NOTE: Do not place your application dependencies here: they belong	
著 🕨 🖿 .idea	15 // in the individual module build.gradle files	
app		
A STC		
gitignore		
a app.imi	18	
a proquard-rules pro	19 Gallprojects {	
Imaudioeffectsettingkit	20 repositories {	
beautysettingkit	21 d flatDir {	
debug	22 dirs 'libs'	
gradle	23 dirs project(':app').file('libs')	
🕨 📷 login		
In trtcaudiocalldemo	25 5	
In trtcliveroomdemo		
Introduction in the second	<pre>26 // maven { url "https://mirrors.tencent.com/nexus/repository/maven_public/" }</pre>	
Introvideocalldemo	27 jcenter()	
Introvoiceroomdemo	28 google()	
.gitignore	29 4 }	
aradle properties	30	
gradie.properties	31	
gradiew bat	32 b dtask clean(tyne, Delete) {	
> du local properties	delate rest Project huildhin	
README.md	adetete rootproject.buitdbir	
settings.gradle	34 0}	
TRTCScenesDemo.iml	35	
g 🕨 🔢 External Libraries	36 ext {	
Scratches and Consoles	37 compileSdkVersion = 25	
E	38 buildToolsVersion = "28.0.3"	
*	39 supportSdkVersion = $"25.4.0"$	
	40 minSdkVersion = 16	
n	11 torgetSkVerien = 26	
S N	42 LILEAVSOK="COM.TENCENT.LILEAVSLITEAVSDA_IKIC:LATEST.FELEASE"	
	43 imSdk = 'com.tencent.imsdk:4.7.10'	
	44 versionCode = 1	
nres	45 versionName = "v1.0"	
Capt	46 ndkAbi = 'armeabi'//,'armeabi-v7a', 'arm64-v8a'	l l l l l l l l l l l l l l l l l l l
out c	47 aekit version = '1.0.10-cloud'	ī
Layc		×P
Eq.		Oret
≠ 9: Version Control 🗵 Terminal 🔨 Build = 6	Logeat III TODO	C Event Log
Gradle sync finished in 19 s 857 ms (6 minutes ago)		3 4 spaces Git: master 🍙 👮

4. app/build.gradleの中に、aar パッケージのコードを追加して引用します。

	TRTCScenesDemo [-/github/TRTCSDK/Android/TRTCScenesDemo] - build.gradie (:app)	
New York Contemporation (Contemporation (Conte	▲ app マ No Devices マ ▶ C: 三 道 15. 00 長 目 Git ビ ✓ ③ つ №	E # L % Q []
동 🗐 Project 👻 😳 😤 💠 —	🔊 build.gradie (app) 🛛	
TRTCScenesDemo in app in build.gradie Project * Image: State in the image: State in th	TRTCSconesDemo[-jglthub/TRTCSDK/Androk(/TRTCScenesDemo]-build gradle (app) TrtCsconesDemo[-jglthub/TRTCSDK/Androk(/TRTCScenesDemo]-build gradle (app) TrtCsconesDemo[-jglthub/TRTCSDK/Androk(/TRTCScenesDemo]-build gradle (app) TrtCsconesDemo[-jglthub/TRTCSDK/Androk(/TRTCSCenesDemo]-build gradle (app) TrtCsconesDemo[-jglthub/TRTCSDK/Androk(/TRTCSCenesDemo]-build gradle (app) TrtCsconesDemo[-jglthub/TRTCSDK/Androk(/TRTCSCenesDemo]-build gradle (app) TrtCsconesDemo[-jglthub/TCommon.so" doNotStrip "*/armeabi/libYTCommon.so" doNotStrip "*/arme64-v8a/libYTCommon.so" doNotStrip "*/arme64-v8a/libYTCommon.so" doNotStrip "*/arme64-v8a/libYTCommon.so"	p3
2 Layout Capitules In 2	<pre>42 43 b dependencies { 44 45 45 46 47 47 47 48 48 49 49 49 49 49 49 49 49 49 49 49 49 49</pre>	Device Tile Explorer
부 운: Version Control 团 Terminal 🔨 Build 프 6: Lo	egent = TODO	C Event Log
Gradle sync finished in 19 s 857 ms (10 minutes apo)	53:1 LF UTF-8 4	spaces Git: master 🔒 👼



5. app/build.gradle的defaultConfigの中で、Appが使用するCPUアーキテクチャを指定します。

```
defaultConfig {
  ndk {
  abiFilters "armeabi", "armeabi-v7a", "arm64-v8a"
  }
}
```

```
説明:
現在 TRTC SDKは、armeabi、armeabi-v7a、arm64-v8aをサポートしています。
```

6. 【Sync Now】をクリックして、TRTC SDK統合のタスクは完了です。

SDKの統合(jar)

aarライブラリを統合したくない場合は、jarおよび soライブラリをインポートする方式で TRTC SDKを統合する ことが可能です。

- 1. 最新バージョンのjar 圧縮パッケージをダウンロード します。ファイルパスは SDK/LiteAVSDK_TRTC_xxx.zip (このうち xxx は TRTC SDKのバージョンナンバー)です。
- 2. 解凍するとlibsディレクトリが取得できます。中には主にjar ファイルと so ファイルフォルダが含まれています。
- 3. 解凍したjar ファイルおよび armeabi、 armeabi-v7a、 arm64-v8a ファイルフォルダを app/libs ディレク トリの下にコピーします。

	📃 libs				
			Q 搜索		
 版本历史.md Android Electron iOS Mac README.md SDK下载.md Web Windows WXMini 	 ■ README.md ■ SDK ■ TRTC说明文档.pdf ■ TRTCScenesDemo ● TRTCScenes运行文档.pdf ■ TRTCSimpleDemo ■ 	 app audioeffectsettingkit beautysettingkit build.gradle debug gradle.properties gradlew gradlew.bat local.properties login README.md settings.gradle trtcaudiocalldemo trtcliveroomdemo TRTCScenesDemo.iml trtcvicecroomdemo 	Q 搜索 app.iml build.gradle fibs proguard-rules.pro fiss fiss fiss fiss fiss fiss fiss fis	arm64-v8a armeabi armeabi-v7a ≩ liteavsdk.jar	•



4. app/build.gradleの中に、jar ライブラリのコードを追加して引用します。



5. app/build.gradleの中に、soライブラリのコードを追加して引用します。

```
sourceSets {
main {
jniLibs.srcDirs = ['libs']
}
```





6. app/build.gradleの defaultConfigの中で、Appが使用するCPUアーキテクチャを指定します。



説明: 現在 TRTC SDKは、armeabi、armeabi-v7a、arm64-v8aをサポートしています。

7. 【Sync Now】をクリックして、TRTC SDK統合のタスクは完了です。

App 権限の設定

AndroidManifest.xml の中で Appの権限を設定します。TRTC SDKには次の権限が必要となります。

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feat

注意:

android:hardwareAccelerated="false" は設定しないでください。ハードウェアアクセラレーションを無 効にすると、相手側のビデオストリームがレンダリングできなくなります。

混同ルールの設定

proguard-rules.pro ファイルで、 TRTC SDK関連のクラスを非混同リストに追加します。

-keep class com.tencent.** { *; }

Appパッケージングのパラメータの設定

app/build.gradleの下に、次の情報を追加します。

packagingOptions {
 pickFirst '**/libc++_shared.so'
 doNotStrip "*/armeabi/libYTCommon.so"
 doNotStrip "*/armeabi-v7a/libYTCommon.so"
 doNotStrip "*/x86/libYTCommon.so"
 doNotStrip "*/arm64-v8a/libYTCommon.so"
}





C++ インターフェースを介してSDKを使用(オプション)

開発にJavaではなくC++ インターフェースを使用したい場合は、この手順を実行できます。Java言語のみを使用 してTRTC SDKを呼び出す場合は、この手順を無視してください。

1.まず、上記の指示に従ってjarおよび soライブラリをインポートする方式で TRTC SDKを統合します。

2.ヘッダーファイルをコピー:SDK中のC++ヘッダーファイルをプロジェクトにコピーし(パ

ス: SDK/LiteAVSDK_TRTC_xxx/libs/include)、CMakeLists.txtでincludeフォルダーパスと soライブラリの動 的リンクを設定します。

```
cmake_minimum_required(VERSION 3.6)
# C++ インターフェースヘッダーファイルパスの設定
include_directories(
${CMAKE_CURRENT_SOURCE_DIR}/include # SDK/LiteAVSDK_TRTC_xxx/libs/includeからコピーする
)
add_library(
native-lib
SHARED
native-lib.cpp)
# libliteavsdk.so動的ライブラリパスの設定
```



add_library(libliteavsdk SHARED IMPORTED) set_target_properties(libliteavsdk PROPERTIES IMPORTED_LOCATION \${CMAKE_CURRENT_SOURCE_DIR}/../.. /../libs/\${ANDROID_ABI}/libliteavsdk.so) find_library(log-lib log) # *libliteavsdk.so 動的リンクの設定* target_link_libraries(native-lib libliteavsdk \${log-lib})

 ネームスペースの利用: C++ のすべてのプラットフォームのインターフェースのメソッド、タイプなどはいず れも trtc ネームスペースで定義されています。コードをより簡潔にするため、trtc ネームスペースを直接使用 することをお勧めします。

using namespace trtc;

説明:

- Android Studio C/C++ 開発環境の設定方法は、Android Studioの公式ドキュメントをご参照ください:プロジェクトへのC/C++コードの追加。
- 現在、TRTCバージョンのSDKのみがC++インターフェースをサポートしています。C++ インターフェ ースの使用方法については、 全プラットフォーム(C++) APIの概要をご参照ください。

Mac

最終更新日::2022-04-02 17:49:38

このドキュメントでは、主にTRTC SDK(Mac)を迅速にプロジェクトに統合する方法を紹介します。以下のステ ップにしたがって設定するだけで、SDK統合のタスクが完了します。

開発環境要件

- Xcode 9.0+。
- OS X10.10+のMac実機。
- プロジェクトに有効な開発者の署名が設定してあること。

TRTC SDKの統合

CocoaPodsを使用して自動でローディングするか、または手動で、先ずSDKをダウンロードして、それを現在の プログラムのプロジェクトにインポートする方式を選択できます。

CocoaPods

1. CocoaPodsのインストール

端末のウィンドウに次のコマンドを入力します(事前にMacにRuby環境をインストールしておく必要があります)。

sudo gem install cocoapods

2. Podfileファイルの作成

プロジェクトが存在するパスに入り、次のコマンドラインを入力するとプロジェクトパスの下にPodfileファイル が現れます。

pod init

3. Podfileファイルの編集

Podfileファイルを編集します。次の2種類の設定方法があります。

• 方式1: Tencent Cloud LiteAV SDKのpodパスを使用します。

🕗 Tencent Cloud

```
platform :osx, '10.10'
```

target 'Your Target' do
pod 'TXLiteAVSDK_TRTC_Mac', :podspec => 'https://liteav.sdk.qcloud.com/pod/liteavsdkspec/TXLit
eAVSDK_TRTC_Mac.podspec'
end

• 方式2: CocoaPodの公式ソースを使用します。バージョンナンバーの選択をサポートしています。

```
platform :osx, '10.10'
source 'https://github.com/CocoaPods/Specs.git'
```

```
target 'Your Target' do
pod 'TXLiteAVSDK_TRTC_Mac'
end
```

4. SDKのインストールおよび更新

端末のウィンドウに次のコマンドを入力して、TRTC SDKのインストールを実行します。

pod install

または次のコマンドを使用してローカルライブラリのバージョンを更新します。

pod update

podコマンドの実行が完了すると、SDKを統合した.xcworkspace という拡張子のプログラムファイルが生成されますので、これをダブルクリックして開きます。

手動による統合

- 1. TRTC-SDK のMacバージョンをダウンロードします。
- 2. お客様のXcodeのプロジェクトを開き、ステップ1でダウンロードしたframeworkプロジェクトにインポート します。



3. 動作させたいtargetを選択し、Build Phasesの項目を選択します。

😣 🗕 🔕 🕨 📄 🚳 TRTC	CDemo 👌 💻 My Mac	TRTCDe	emo	Build TRTCDemo: Suc	ceeded	2018/12/21 at 3:09 PM		6	{}
	器 < > 🖹 TR								< 🛆 >
	General	Capabiliti	es	Resource Tags	Info	Build Settings	Build Phases	Build	l Rules
 TRTCDemo Products 	PROJECT					🗐 Filter			
Frameworks	A TRTCDemo			Target Dependencie	e (O iter	ne)			
	TARGETS			larger Dependencie	S (O ILEI	115/			
	🙆 TRTCDemo)	►	Compile Sources (7	items)				
				Link Binary With Lib	raries (3	3 items)			
			Þ	Copy Bundle Resour	rces (5 it	tems)			

4. Link Binary with Librariesの項目をクリックして展開し、一番下の「+」アイコンをクリックして依存ラ イブラリを追加します。

	CDemo 〉 🛄 My Mac							▲ 6 { }
	🔡 < 🚿 🖻 TRT							$\langle \Delta \rangle$
🔻 <u>À</u> TRTCDemo	General	Capabiliti	es Resource Tag	s Info	Build S	ettings	Build Phases	Build Rules
	PROJECT					🕞 Filter		
Frameworks	TARGETS		Target Depende	encies (0 iter	ms)			
	🙆 TRTCDemo		Compile Source					
			▼ Link Binary Wit	h Libraries ((0 items)			
				Name			Status	
					Add fra	ameworks 8	k libraries here	
				<u>+</u> –	Drag	to reorder	frameworks	
			Copy Bundle Re	sources (5 i	tems)			×

- 5. 順番に、ダウンロードしたSDK Frameworkおよびそれに必要な依存ライブラ
 - リ: AudioUnit.framework 、 libc++.tbd 、 Accelerate.framework を追加します。



追加した後は以下の通りです。

🗧 🍋 🌒 下 🔲 🙆 TRTCDemo 👌	🜉 My Mac 🕺 🕺 Indexing					
E R Q A O E D F	踞 < > 🖹 TRTCDemo					Ð
🔻 🖹 TRTCDemo 🛛 M	📘 General 🛛 Signing &	Capabilities Reso	ource Tags Info	Build Settings	Build Phases	Build R
TRTCDemo Troducts Frameworks Accelerate framework	PROJECT TRTCDemo TARGETS	+ ► Dependencies	s (O items)	Filter		
 AudioUnit.framework libc++.tbd TXLiteAVSDK_TRTC_Mac.framework 	🙆 TRTCDemo	► Compile Source	ces (12 items) ith Libraries (4 items)			×
			Name		Status	
			🚔 AudioUnit.framewo	rk	Required 🗘	
			🚔 Accelerate.framewo	ork	Required 🗘	
			📝 libc++.tbd		Required 🗘	
			TXLiteAVSDK_TRT	C_Mac.framework	Required 🗘	
			+ — Dra	g to reorder linked bi	naries	
		Copy Bundle F	Resources (10 items)			×

カメラとマイクの使用権限の許可

SDKの音声ビデオ機能を使用するには、マイクとカメラの使用権限を許可する必要がありますので、Appの Info.plistの中に次の2項目を追加します。システムが使用許可のダイヤログボックスをポップアップするときに 表示されるマイクとカメラの情報にそれぞれ対応します。

- Privacy Microphone Usage Description、さらにマイク使用目的のプロンプトを記入します。
- Privacy Camera Usage Description、さらにカメラ使用目的のプロンプトを記入します。
 以下の通りです。

CDemo 〉 My Mac			d TRTCDemo: Succee	eded 20		3:09 PM		<u>6</u>	
TRTCDemo.x									
멾 < > 📓 TRTCDemo									
	General	Capabilities	Resource Tags		Build Se	ettings	Build Phases	Build Rule	s
PROJECT	- 0	Anniination T							
A TRTCDemo	 Custom macOs 	Application la	arget Properties						
TARGETS		Кеу			Туре	Value			
		Bundle vers	sions string, short	0	String	1.0			
		Privacy - M	icrophone Usage Des	cript 🖒	String				
		Privacy - C	amera Usage Descrip.	. 000		0		_	
		Bundle ider	ntifier	\$		\$(PRO	DUCT_BUNDLE_ID	ENTIFIER)	
		Main story	ooard file base name	\$		Main			
		InfoDiction	ary version	\$		6.0			
		Bundle vers	sion	\$		1			
		Executable	file	٢		\$(EXEC	CUTABLE_NAME)		
		Principal cl	ass	\$		NSApp	lication		
		Bundle OS	Type code	\$		APPL			
		Icon file		\$					
		Minimum s	stem version	\$		\$(MAC	OSX_DEPLOYMEN	IT_TARGET)	
		Localizatio	n native development i	region 🖒		\$(DEVI	ELOPMENT_LANG	JAGE)	\$
		Copyright (human-readable)	٢		Copyri	ght © 2018 rusha	nting. All right	s reserved.
		Bundle nan	ne	\$		\$(PRO	DUCT_NAME)		

Appで**App Sandbox**または**Hardened Runtime**を有効にしている場合は、 Network 、 Camera 、 Audio Input の選択項目にチェックを入れる必要があります。

• App Sandboxの設定は以下の通りです。

▼ [́́͡∰] App Sandbox		×
Network	 Incoming Connections (Server) Outgoing Connections (Client) 	
Hardware	 Camera Audio Input USB Printing Bluetooth 	
App Data	Contacts Location Calendar	

• Hardened Runtimeの設定は以下の通りです。

▼ ◯ Hardened Runtime		×
Runtime Exceptions	Allow Execution of JIT-compiled Code Useful in conjunction with JavaScriptCore.framework or other frameworks relying on JIT compilation. Allows creating writable and executable memory using the MAP_JIT flag.	
	Allow Unsigned Executable Memory Useful for legacy applications that create executable code in memory. Allows creating writable and executable memory without using the MAP_JIT flag.	
	Allow DYLD Environment Variables Allows an application to be impacted by DYLD environment variables, which can be used to inject code into the process.	
	Disable Library Validation Allows an application to load plug-ins or frameworks signed by other developers.	
	Disable Executable Memory Protection Disables all code signing protections on the application while executing. Useful for legacy applications that modify their own executable code in memory.	
	Debugging Tool Declares the application as a debugger. Useful for applications that need to attach to other processes or get task ports.	
Resource Access 🔽	Audio Input Allows recording of audio using the built-in microphone, if available, along with access to audio input using any Core Audio API that supports audio input.	
	Allows capture of movies and still images using the built-in camera, if available.	
	Location Grants access to Location Services location information.	
	Contacts Provides read/write access to contacts in the user's address book; allows apps to infer the default address book if more than one is present on a system.	
	Calendar Provides read/write access to the user's calendars.	
	Photos Library Provides read/write access to the user's Photos library.	
	Apple Events Allows posting of AppleEvents to other applications.	

TRTC SDKの引用

TRTC SDKでは2種類の呼び出し方式をサポートしていますので、いずれかをお選びください。

方式1: Objective-CまたはSwiftインターフェースによるTRTC SDKの引用

Objective-CまたはSwiftコードの中でSDKを使用する方式は2種類あります。

• **モジュールの引用**:プロジェクトのSDK APIを使用したいファイルの中に、モジュールを追加して引用しま す。

@import TXLiteAVSDK_TRTC_Mac;

 ヘッダーファイルの引用:プロジェクトのSDK APIを使用したいファイルの中に、具体的なヘッダーファイル をインポートします。

#import TXLiteAVSDK_TRTC_Mac/TRTCCloud.h

方式2:C++インターフェースによるTRTC SDKの引用

 ヘッダーファイルの引用: C++インターフェースを使用してMacアプリケーションを開発したい場合は、 TXLiteAVSDK_TRTC_Mac. framework/Headers/cpp_interface ポートしてください。

#include TXLiteAVSDK_TRTC_Mac/cpp_interface/ITRTCCloud.h

 ネームスペースの利用: C++の全プラットフォームのインターフェースのメソッド、タイプなどはいずれも trtcネームスペースの中に定義されています。コードをより簡潔にするため、trtcネームスペースを直接使用す ることをお勧めします。

using namespace trtc;

説明:

C++インターフェースの使用方法については、全プラットフォーム(C++)APIの概要をご参照ください。

Windows

最終更新日::2021-04-12 15:37:31

このドキュメントでは、Tencent Cloud TRTC SDK(Windows C#およびC++ 版をプロジェクトに迅速に統合 する方法を紹介します。

開発環境要件

- OS:Windows 7およびそれ以上のバージョン。
- 開発環境: Visual Studio 2010およびそれ以上のバージョン、Visual Studio 2015の使用を推奨します。
- 開発アーキテクチャ:.Net Framework 4.0およびそれ以上のバージョン。

TRTC C# SDKの統合

このセグメントでは、例として簡単な Winformのプロジェクトを作成し、Visual Studioのプログラムの中でC# SDKを統合する方法を紹介します。

ステップ1: Windows SDKのダウンロード

SDKをダウンロード し、ファイルを解凍して開きます。これには次の部分が含まれています。

ディレクトリ名	説明
xxxDemo	C++ Demo ソースコードおよびC# Demo ソースコード
CPlusPlus	C++版32ビット/64ビットの依存する SDK ライブラリファイル
CSharp	C#版32ビット/64ビットの依存するSDK ライブラリファイル

ここの例では、SDK ディレクトリの下のC#版のSDK ファイルを引用するだけで済みます。

ステップ2:プログラムの新規作成

Visual Studioを開き、 TRTCCSharpDemo という名前の Winform アプリケーションプログラムを新規作成します。

ステップ3:ファイルのコピー

解凍後のSDK ファイルフォルダを TRTCCSharpDemo.csproj が存在するディレクトリにコピーします。

() 説明:

C# SDKのみが必要な場合は、SDKのパスの下の CPlusPlus ディレクトリを削除して構いません。

ステップ4:プログラム設定の修正

ステップ4.1:追加および引用

- 1. Visual Studio の【生成】ディレクトリの下から【Configuration Manager】を見つけて開きます。
- 2. 【イベントソリューションプラットフォーム】のプルダウン・リストから【新規作成】を選択すると、【ソリ ューションプラットフォームの新規作成】のダイアログボックスがポップアップします。
- 3. 新しいプラットフォームを入力または選択し、【確定】をクリックします。
- 4. 実際のニーズにもとづき、ステップ2-ステップ3を繰り返して、サポートする必要があるソリューションプラットフォームを新規作成します。
- 5. TRTCCSharpDemo プロジェクトが存在するファイルフォルダを開き、テキストエディタを使っ て TRTCCSharpDemo.csproj ファイルを編集します。
- 6. TRTCCSharpDemo.csproj のファイルの中のタグ <itemGroup> の下に次の内容を追加します。

```
//それぞれのプラットフォームの下に引用を追加します
<Reference Include="ManageLiteAV" Condition="'$(Platform)' == 'x64'">
<HintPath>SDK¥CSharp¥Win64¥lib¥ManageLiteAV.dll</HintPath>
</Reference>
<Reference Include="ManageLiteAV" Condition="'$(Platform)' == 'AnyCPU'">
<HintPath>SDK¥CSharp¥Win64¥lib¥ManageLiteAV.dll</HintPath>
</Reference>
<Reference Include="ManageLiteAV" Condition="'$(Platform)' == 'x86'">
<HintPath>SDK¥CSharp¥Win64¥lib¥ManageLiteAV.dll</HintPath>
</Reference>
<Reference Include="ManageLiteAV" Condition="'$(Platform)' == 'x86'">
<HintPath>SDK¥CSharp¥Win64¥lib¥ManageLiteAV.dll</HintPath>
</Reference>
```

ステップ4.2: copy コマンドの追加

- 1. TRTCCSharpDemoの属性のページを開きます。【ソリューションのResource Manager】> 【TRTCCSharpDemo プログラムの右クリックメニュー】>【属性】を選択します。
- 【イベントの生成】>【後続のイベント生成コマンドライン】の中に次のコマンドを追加し、編集完了後に、 様々なプラットフォーム下の SDKの .dll ファイルが自動的にプログラムの実行ディレクトリの下にコピーされ るようにします。次の図のとおりです。



```
set Platform=Win64
SETLOCAL ENABLEDELAYEDEXPANSION
if $(PlatformName)==x86 (
set Platform=Win32
)
copy /Y "$(ProjectDir)SDK¥CSharp¥!Platform!¥lib¥*.dll" "$(ProjectDir)$(OutDir)"
ENDLOCAL
```

ステップ4.3:デバッグ環境の修正

TRTCDemoの属性のページを開き、【生成】を選択して、【プラットフォーム(M)】とトップメニューバーの中のソリューションプラットフォームの設定を一致させます。次の図のとおりです。

ステップ5: SDK バージョンナンバーのプリント

- 1. Form1.csの設計ツールの中にlabelコントロールを追加します。次の図のとおりです。
- 2. Form1.csのコードファイルを開き、次のコードを追加します。

```
using System.Windows.Forms;
using ManageLiteAV; // 1..ネームスペースの引用を追加します
namespace TRTCCSharpDemo
{
public partial class Form1 : Form
{
public Form1()
{
InitializeComponent();
// 2.ITRTCCloud インスタンスを取得し、SDK バージョンナンバーをプリントします
ITRTCCloud lTRTCCloud = ITRTCCloud.getTRTCShareInstance();
this.label1.Text = "SDK version : " + LTRTCCloud.getSDKVersion();
// 3. 使用終了時に、ITRTCCloud インスタンスを手動で破棄する必要があります
ITRTCCloud.destroyTRTCShareInstance();
}
}
}
```

3. F5を押して実行し、SDKのバージョンナンバーをプリントします。次の図のとおりです。

Form1		
	SDK version : 6.7.0.7732	

TRTC C++ SDKの統合

このセグメントでは、簡単なMFC プロジェクトを作成し、Visual Studio プログラムの中で C++ SDKを統合す る方法を紹介します。

ステップ**1**: **SDK**のダウンロード

SDKをダウンロードし、解凍して開きます。これには次の部分が含まれています。

ディレクトリ名	説明
include	詳細なインターフェースの説明がついたAPI ヘッダーファイル
lib	編集用の.lib ファイルおよび実行時にローディングする.dll ファイル

ステップ2: プログラムの新規作成

Visual Studioを開き、TRTCDemoという名前のMFC アプリケーションプログラムを作成します。次の図のとおりです。

迅速に統合する方法を紹介しやすいように、ガイドの**アプリケーションプログラムのタイプ**の画面では、比較的 簡単な**ダイアログベース**のタイプを選択しています。次の図のとおりです。

その他のガイドの設定は、デフォルトの設定を選択してください。

ステップ3:ファイルのコピー

解凍後のLiteAVSDK ファイルフォルダを TRTCDemo.vcxprojが存在するディレクトリの下にコピーします。次の図のとおりです。

ステップ4:プログラム設定の修正

TRTCDemoの属性のページを開きます。【ソリューションのResource Manager】 >【TRTCDemo プログラムの右クリックメニュー】>【属性】と進みます。次のステップにしたがって設定してください。

1. Includeのディレクトリの追加:

【C/C++】>【Genaral】>【Include ディレクトリの追加】で、SDK ヘッダーファイルディレクトリ \$(ProjectDir)LiteAVSDK¥include と \$(ProjectDir)LiteAVSDK¥include¥TRTC を追加します。次の図のとお りです。

2. ライブラリのディレクトリの追加:

【リンケージ】>【通常】>【ライブラリディレクトリの追加】で、SDKライブラリディレクトリ \$(ProjectDir)LiteAVSDK¥lib を追加します。次の図のとおりです。

3. ライブラリファイルの追加:

【リンケージ】>【入力】>【依存項目の追加】で、 SDK ライブラリファイル liteav.lib を追加します。 次の図のとおりです。

4. copy コマンドの追加:

【イベントの生成】>【後続のイベントの生成】>【コマンドライン】で、コマンド copy /Y "\$(ProjectDir)LiteAVSDK¥lib¥¥¥*.dll" "¥\$(OutDir)" のコピーを追加します。編集が完了すると、自動で SDKの.dll ファイルがプログラムの実行ディレクトリの下にコピーされます。次の図のとおりです。

ステップ5:SDK バージョンナンバーのプリント

• CTRTCDemoDlg::OnInitDialog 関数の中に、以下のテストコードを追加します。

• C++

CWnd *pStatic = GetDlgltem(IDC_STATIC); pStatic->SetWindowTextW(szText); :::

• F5キーを押して実行すると、SDKのバージョンナンバーがプリントされます。次の図のとおりです。

RTCDemo			
	SDK version: 5.9.0.6932		
		确定	取消

よくあるご質問

• 次のエラーが出現した場合は、 プログラム設定の修正にしたがって、SDKの引用がプログラムの中に引用され ているかチェックしてください。

エラー CS0246は、タイプまたはネームスペース "ManageLiteAV" が見つからない場合です(**using** のコマ ンドまたはプログラムセットの引用が欠如していませんか?)

 次のエラーが生じた場合は、プログラム設定の修正にしたがって、プログラムの動作プラットフォーム環境が ソリューションの現在のターゲットプラットフォームに修正されているかチェックしてください。

System.BadImageFormatException: "ファイルまたはプログラムセット "ManageLiteAV, Version=2.0.715 2.18518, Culture=neutral, PublicKeyToken=null" またはそれの或る依存項目がローディングできません。形式が正しくないプログラムをローディングしようとしています。" 次のエラーが生じた場合は、プログラム設定の修正にしたがって、生成したイベントが実行ディレクトリの中に正しく追加されているかチェックしてください。

System.IO.FileNotFoundException: "ファイルまたはプログラムセット "ManageLiteAV.dll" またはそれ の或る依存項目がローディングできません。指定したコンポーネントが見つかりません。"

- Windows のバージョンの違いにより互換性の問題が存在している可能性があります。現在 C# SDK の中には 互換性の問題を解決する dll ファイルを新規追加しています。ファイルリストは次の図のとおりです。
- 次のエラーが生じた場合は、前述のプログラム設定にしたがって、SDKヘッダーファイルのディレクトリが正しく追加されているかチェックしてください。

fatal **error** C1083: **include** ファイルを開くことができません: "TRTCCloud.**h**": **No** such **file** or di rectory

次のエラーが生じた場合は、前述のプログラム設定にしたがって、SDKライブラリのディレクトリとライブラリファイルが正しく追加されているかチェックしてください。

error LNK2019: 解析できない外部シンボル "_declspec(dllimport) public: static class TXString _ _cdecl TRTCCloud::getSDKVersion(void)" (__imp_?getSDKVersion@TRTCCloud@@SA?AVTXString@@XZ)、こ の記号が関数 "protected: virtual int __thiscall CTRTCDemoDlg::OnInitDialog(void)" (?OnInitDial og@CTRTCDemoDlg@@MAEHXZ) の中に引用されています

Web

最終更新日:: 2022-02-22 15:45:52

ここでは、主にTencent Cloud TRTC Web SDK をプロジェクトに素早く統合する方法を紹介します。

サポートするプラットフォーム

WebRTCのテクノロジーはGoogleが初めて提唱し、Chrome、Edge、Firefox、Safari、Operaなどのブラウザ ですべてサポートされています。Tencent Cloud TRTC Web SDKはWebRTCパッケージをベースにしたもので す。Tencent Cloud TRTC Web SDKのサポートの詳細については、サポートするプラットフォームをご参照くだ さい。

 ユーザーのユースケースが主に教育シーンである場合は、教師用端末では安定性がより優れたElectron ソリュ ーションの使用をお勧めし、大小2チャネル画面、よりフレキシブルなスクリーンシェアリング方法およびより 強力な弱ネットワークリカバリー能力をサポートしています。

Tencent Cloud TRTC Web SDKの詳細なサポートレベルの表については、サポートしているプラットフォームを ご参照ください。

注意:

- ブラウザでTRTC Web SDK機能テスト画面を開けば、現在のブラウザがWebRTCのすべての機能をサポートしているかどうかチェックすることができます。例:WebViewなどのブラウザ環境。
- H.264の著作権上の制限により、Huawei Chrome 88 より前のバージョンは H264 エンコーディン グを使用できません(つまり、ストリームをプッシュできません)。HuaweiデバイスのChromeブラウ ザで TRTC Web SDK を使用してストリームをプッシュするには、チケットを提出し、VP8コーデック の有効化を申請してください。

URLドメイン名プロトコルの制限

ユースケース	プロトコル	受信 (再生)	送信(マイク・ オン)	画面共有	備考
本番環境	HTTPSプロトコル	サポートあ り	サポートあり	サポートあ り	推奨

ユースケース	プロトコル	受信 (再生)	送信(マイク・ オン)	画面共有	備考
本番環境	HTTPプロトコル	サポートあ り	サポートなし	サポートな し	
ローカル開発環 境	http://localhost	サポートあ り	サポートあり	サポートあ り	推奨
ローカル開発環 境	http://127.0.0.1	サポートあ り	サポートあり	サポートあ り	
ローカル開発環 境	http://[ローカルマシン IP]	サポートあ り	サポートなし	サポートな し	
ローカル開発環 境	file:///	サポートあ り	サポートあり	サポートあ り	

ファイアウォールの制限

TRTC Web SDKは、以下のポートに依存してデータ伝送を実施しますので、ファイアウォールのホワイトリスト に追加してください。

- TCPポート:8687
- UDPポート:8000、8080、8800、843、443、16285
- ドメイン名:qcloud.rtc.qq.com

TRTC Web SDKの統合

NPM統合

1. プロジェクトにSDKパッケージをインストールするには、npmを使用する必要があります。

npm install trtc-js-sdk --save

2. プロジェクトのスクリプトでモジュールを導入します。

import TRTC from 'trtc-js-sdk';
Script統合

下記のコードをWebページに追加するだけで完了です。

<script src="trtc.js"></script>

関連リソース

SDKダウンロードアドレス:クリックしてダウンロード。

初期化フローおよびAPIの使用法の詳細については、以下のガイドをご参照ください。

機能	Sample Codeガイド
基本的なオーディオビデオ通話	ガイドリンク
インタラクティブライブストリーミング	ガイドリンク
カメラおよびマイクの切り替え	ガイドリンク
ローカルビデオのプロパティの設定	ガイドリンク
ローカルオーディオまたはビデオの動的な停止と開始	ガイドリンク
画面共有	ガイドリンク
音量計測	ガイドリンク
ユーザー定義キャプチャとカスタマイズ再生レンダリング	ガイドリンク
ルーム内アップリンクユーザー数の制限	ガイドリンク
バックグラウンドミュージックと効果音の実装ソリューション	ガイドリンク
通話前の環境およびデバイステスト	ガイドリンク
通話前のネットワーク品質テスト	ガイドリンク
デバイス挿抜動作チェック	ガイドリンク
CDNへのプッシュの実現	ガイドリンク
ビッグスモールストリームの伝送を有効にする	ガイドリンク
美顔を有効にする	ガイドリンク
ウォーターマークを有効にする	ガイドリンク
ルーム間マイク接続の実現	ガイドリンク



說明:

その他の機能についてはクリックして確認してください。

Electron

最終更新日:: 2022-01-19 15:22:54

ここでは、主にTencent Cloud TRTC Electron SDK をプロジェクトに素早く統合する方法を紹介します。

サポートするプラットフォーム

- Windows(PC)
- Mac

TRYC Electron SDKの統合

手順1:Node.jsのインストール

• Windowsプラットフォームのインストールガイド

• MacOSプラットフォームのインストールガイド

1. WindowsOSに従って、最新バージョンのNode.js インストールパック Windows Installer (.msi) 64bit を選択、ダウンロードします。

アプリケーションプログラムリストにあるNode.js command promptを開き、コマンドラインのポートを起動し、その後の手順における各コマンドの入力に使用します。



手順2: Electronのインストール

コマンドラインポートで以下のコマンドを実行し、Electronをインストールします。バージョンは>=4.0.0を推 奨します。

\$ npm install electron@latest --save-dev

手順3: Electronバージョン TRTC SDKのインストール

1. Electron項目でnpmコマンドを使用してSDKパッケージをインストールします:

```
$ npm install trtc-electron-sdk@latest --save
```

説明: TRTC Electron SDK最新版では、 <mark>trtc-electron-sdk</mark> でクエリーできます。

2. プロジェクトのスクリプトでモジュールを導入して、使用します。

```
const TRTCCloud = require('trtc-electron-sdk').default;
// import TRTCCloud from 'trtc-electron-sdk';
this.rtcCloud = new TRTCCloud();
// SDKバージョン番号の取得
this.rtcCloud.getSDKVersion();
```

v7.9.348から、TRTC Electron SDKはtrtc.d.tsファイルを増加しており、 TypeScriptを使用する開発者には便利になりました:

```
import TRTCCloud from 'trtc-electron-sdk';
const rtcCloud: TRTCCloud = new TRTCCloud();
// SDKバージョン番号の取得
rtcCloud.getSDKVersion();
```

パッケージで実行可能なプログラム

手順1:パッケージングツールのインストール

 パッケージングツール electron-builder を使用してパッケージングすることを推奨します。以下のコマン ドを実行して electron-builder をインストールできます。

\$ npm install electron-builder@latest --save-dev

2. Electronバージョンの TRTC SDK(すなわち trtc_electron_sdk.node ファイル)を正しくパッケージする ために、以下のコマンドを実行して native-ext-loader ツールもインストールする必要があります。

\$ npm install native-ext-loader@latest --save-dev

手順2:webpack.config.js設定の修正

webpack.config.js は、項目アーキテクチャの設定情報を含んでいます。 webpack.config.js ファイルの位置 は以下のとおりです。

- 通常、 webpack.config.js は項目のルートディレクトリにあります。
- create-react-app 使用して項目を新規作成する状況では、この設定ファイルは node_modules/reactscripts/config/webpack.config.js です。
- vue-cli を使用して項目を新規作成する状況では、webpackの設定は vue.config.js 設定の configureWebpack のプロパティにあります。
- プロジェクトファイルがカスタマイズされている場合は、ご自身でwebpack設定を検索してください。
- 初めに webpack.config.js を構築するときは、受信名を --target_platform のコマンドラインパラメータ にできるため、コードの構築プロセスで異なる目標プラットフォームの特徴に応じて正しくパッケージできま す。 module.exports の前に以下のコードを追加します。

```
const os = require('os');
const targetPlatform = (function(){
let target = os.platform();
for (let i=0; i<process.argv.length; i++) {
if (process.argv[i].includes('--target_platform=')) {
target = process.argv[i].replace('--target_platform=', '');
break;
}
}
if (!['win32', 'darwin'].includes) target = os.platform();
return target;
})();
```

注意:

os.platform() によって返される結果では、"darwin" は Macのプラットフォームを表します。64ビットまたは 32ビットに関わらず、"win32" は Windowsプラットフォームを表します。

 その後 rules オプションでは以下の設定を追加します。 targetPlatform 変数は rewritePath を使って、 異なる目標プラットフォームに従って違う設定に切り替えることができます。

```
rules: [
{
  test: /¥.node$/,
  loader: 'native-ext-loader',
  options: {
  rewritePath: targetPlatform === 'win32' ? './resources' : '../Resources'
  }
},
]
```

この設定の意味は以下のとおりです。

- Windows \mathcal{O} . exe $\mathcal{O}_{\mathcal{P}}$ $\mathcal{O}_$
- Macの .dmg をパッケージするときは、 native-ext-loader に [アプリケーションディレクト リ]/Contents/Frameworsk/../Resources ディレクトリでTRTC SDKをロードします。

package.json のスクリプト構築で、 --target_platform パラメータを追加するには、以下を実行します。

手順3:package.json設定の修正

package.json は項目のルートディレクトリにあり、そのうち項目のパッケージに必要な情報を含みます。しかし、デフォルトの状況では、 package.json の中のパスを修正しなければ順調にパッケージできません。以下の 手順に従ってこのファイルを修正できます。

1. main 設定の修正。

```
// 多くの状況では、mainファイル名は任意に設定できます。例えば、 TRTCSimpleDemoのものは次のように
設定できます。
"main": "main.electron.js",
```

// しかし、 create-react-app のスキャフォールディングを使用して新規作成した項目、mainファイルは次 のように設定する必要があります: "main": "public/electron.js",

2. 以下の `build` 設定をコピーし、`package.json` ファイルに追加します。これは `electron-builder` が読み 取る必要のある設定情報です。

```
"build": {
"appId": "[appId はご自身で定義してください]",
"directories": {
"output": "./bin"
},
"win": {
"extraFiles": [
```

```
"from": "node_modules/trtc-electron-sdk/build/Release/",
"to": "./resources",
"filter": ["**/*"]
}
]
},
"mac": {
"extraFiles": [
{
"from": "node_modules/trtc-electron-sdk/build/Release/trtc_electron_sdk.node",
"to": "./Resources"
}
]
}
```

3. `scripts`ノードにおいて以下のアーキテクチャおよびパッケージしたコマンドスクリプトを追加します。 こ こでは、`create-react-app`および`vue-cli`項目を例に、その他のツールで作成した項目もこの設定を参考に することができます。

```
// create-react-app項目にはこの設定を使用してください
"scripts": {
"build:mac": "react-scripts build --target_platform=darwin",
"build:win": "react-scripts build --target platform=win32",
"compile:mac": "node_modules/.bin/electron-builder --mac",
"compile:win64": "node_modules/.bin/electron-builder --win --x64",
"pack:mac": "npm run build:mac && npm run compile:mac",
"pack:win64": "npm run build:win && npm run compile:win64"
}
// vue-cli項目にはこの設定を使用してください
"scripts": {
"build:mac": "vue-cli-service build --target platform=darwin",
"build:win": "vue-cli-service build --target_platform=win32",
"compile:mac": "node modules/.bin/electron-builder --mac",
"compile:win64": "node modules/.bin/electron-builder --win --x64",
"pack:mac": "npm run build:mac && npm run compile:mac",
"pack:win64": "npm run build:win && npm run compile:win64"
}
```

パラメータ	説明
main	Electron のエントリーファイルは、一般には自由に設定できます。しか し、項目を create-react-app のスキャフォールディングを使用して作 成した場合は、エントリーファイルは public/electron.js に設定する 必要があります



build.win.extraFiles	Windowsプログラムをパッケージする場合は、 electron- builder は from が示すディレクトリ下のすべてのファイルをbin/win- unpacked/resources(すべて小文字表記)にコピーします
build.mac.extraFiles	Macプログラムをパッケージする場合は、 electron-builder は from が 示す trtc_electron_sdk.node ファイルをbin/mac/your-app- name.app/Contents/Resources(頭文字は大文字表記)にコピーしま す
build.directories.output	パッケージファイルの出力パス。例えば、この設定を bin ディレクトリ に出力する場合は、実際のニーズに従って修正します
build.scripts.build:mac	Macプラットフォームを目標にスクリプトを構築します
build.scripts.build:win	Windowsプラットフォームを目標にスクリプトを構築します
build.scripts.compile:mac	Macの .dmgインストールファイルをコンパイルします
build.scripts.compile:win64	Windowsの .exeインストールファイルをコンパイルします
build.scripts.pack:mac	まず build:macアーキテクチャコードをコールしてからcompile:macを コールして .dmgインストールファイルをパッケージします
build.scripts.pack:win64	まず build:winアーキテクチャコードをコールしてからcompile:win64を コールして .exeインストールファイルをパッケージします

手順4:パッケージコマンドの実行

• Mac .dmgインストールファイルのパッケージ:

\$ cd [項目ディレクトリ] \$ npm run pack:mac

実行に成功すると、パッケージツールは bin/your-app-name-0.1.0.dmg インストールファイルを新規作成しま すので、このファイルのリリースを選択してください。

• Windows .exeインストールファイルのパッケージ:

\$ cd [項目ディレクトリ] \$ npm run pack:win64 実行に成功すると、パッケージツールは bin/your-app-name Setup 0.1.0.exe インストールファイルを新規作 成しますので、このファイルのリリースを選択してください。

注意:

TRTC Electron SDK は、プラットフォームパッケージ(Mac下でWindowsをパッケージする.exe ファイル、またはWindowsプラットフォーム下でMacをパッケージする .dmgファイルなど)を一時的にサポートしません。現在プラットフォームを跨がるパッケージ方法を検討中ですのでご期待ください。

よくあるご質問

1. ファイアウォールにはどのような制限がありますか。

SDKがUDPプロトコルを使用してオーディオ・ビデオ伝送を行っているため、UDPをブロックするオフィスネットワークでは使用できません。類似した問題がおありの際は、企業ファイアウォール制限の対応をご参照の上、 問題及び原因解決にお役立てください。

2. Electronのインストールまたはパッケージ化のトラブル

 Electron統合中にトラブルが生じた場合、例えばインストールのタイムアウトまたは失敗、パッケージ後に trtc_electron_sdk.nodeファイルのロード失敗などの状況が生じた場合は、お問い合わせまでご連絡いただけ ればご質問にお答えします。

参考ドキュメント

- SDK APIマニュアル
- SDK更新ログ
- Simple Demoソースコード
- API Exampleソースコード
- Electronについてのよくあるご質問

クイックインテグレーション(Flutter)

最終更新日:::2022-04-06 17:11:35

ここでは、主にTencent CloudのTRTC SDK(Flutter)をプロジェクトに素早く統合する方法を紹介します。以 下の手順に従って設定すれば、SDKの統合プロセスを完了できます。

注意:

現在、Flutter SDKはAndroidとiOSのみ対応しています。

環境要件

- Flutter 2.0以降のバージョン。
- Androidの開発:
 - 。 Android Studio 3.5以降のバージョン。
 - 。 AppにはAndroid 4.1以降のバージョンのデバイスが必要です。
- iOSの開発:
 - 。 Xcode 11.0以降のバージョン。
 - 。 プロジェクトが有効な開発者による署名を設定済みであることを確認してください。

SDKの統合

Flutter SDKはpubライブラリに公開されています。 pubspec. yaml を設定することで、更新を自動的にダウンロ ードできます。

1. プロジェクトの pubspec.yaml に次の依存関係を記述します:

dependencies: tencent_trtc_cloud: 最新バージョン番号

2. カメラとマイクの許可を有効にすると、音声通話機能が起動します。

iOS端末

1. Info.plist にカメラとマイクの許可申請を追加する必要があります:



<key>NSCameraUsageDescription</key> <string>通常のビデオ通話が行えるよう、カメラの権限を付与する必要があります</string> <key>NSMicrophoneUsageDescription</key> <string>通常の音声通話が行えるよう、マイクの権限を付与する必要があります</string>

2. フィールド io.flutter.embedded_views_preview を追加し、値をYESに設定します。

Android端末

- 1. /android/app/src/main/AndroidManifest.xml ファイルを開きます。
- 2. xmlns:tools="http://schemas.android.com/tools" をmanifestの中に追加します。
- 3. `tools:replace="android:label"をapplicationの中に追加します。

説明: このステップを実行しないと、Android Manifest merge failed コンパイルエラーという問題が発生し ます。

android > a	app > src > main > 🔊 AndroidManifest.xml
1 <m< td=""><td>anifest xmlns:android="<u>http://schemas.android.com/apk/res/android</u>"</td></m<>	anifest xmlns:android=" <u>http://schemas.android.com/apk/res/android</u> "
2	<pre>xmlns:tools="http://schemas.android.com/tools"</pre>
3	<pre>package="com.example.mlp"></pre>
4	io.flutter.app.FlutterApplication is an android.app.Application that</td
5	calls FlutterMain.startInitialization(this); in its onCreate method.
6	In most cases you can leave this as-is, but you if you want to provide
7	additional functionality it is fine to subclass or reimplement
8	FlutterApplication and put your custom class here>
9	<application< td=""></application<>
10	<pre>tools:replace="android:label"</pre>
11	<pre>android:name="io.flutter.app.FlutterApplication"</pre>
12	android:label="mlp"
13	android:icon="@mipmap/ic_launcher">

よくあるご質問

- iOSでパッケージ作成、実行時にCrashが発生した場合はどうすればよいですか?
- iOSでビデオを表示できない(Androidは正常)場合はどうすればよいですか?
- SDKバージョンの更新後、iOS CocoaPodsの実行時にエラーが発生した場合はどうすればよいですか?

- Android Manifest merge failedというコンパイルエラーが発生した場合はどうすればよいですか?
- 署名がないことで、実機でのデバッグでエラーが発生した場合はどうすればよいですか?
- プラグインのswiftファイルを追加または削除した後、buildする際に対応するファイルが見つからない場合は どうすればよいですか?
- 実行エラー"Info.plit, error: No value at that key path or invalid key path: NSBonjourServices"が発生 した場合はどう対処すればよいですか?
- Pod installエラーが発生した場合はどう対処すればよいですか?
- 実行時にiOSのバージョンに依存するエラーが発生した場合はどうすればよいですか?

クイックインテグレーション(QT)

最終更新日::2021-07-29 18:39:10

ここでは、主にTencent Cloud TRTC SDK(QTのMacバージョンおよびWindowsバージョン)をプロジェクトに 素早く統合する方法を紹介します。以下の手順に従って設定すれば、SDKの統合作業を素早く完了できます。

Mac端末統合

開発環境要件

- オペレーティングシステム: Mac10.10以上のバージョン。
- 開発環境: Qt Creator 4.10.3以上のバージョン。Qt Creator 4.13.3以上のバージョンの使用を推奨。
- 開発フレームワーク: Based on Qt 5.10以上。

操作手順

ここではゼロからの簡単なQTTest項目の作成を例に、Qt CreatorプロジェクトでC++クロスプラットフォーム SDKを統合する方法をご紹介します。

手順1: C++クロスプラットフォームSDKのダウンロード

1. SDKをダウンロードし、ファイルを解凍して開きます。

QTTestと同じクラスのディレクトリ下で空のSDKフォルダを作成し、手順1でダウンロードし
 た TXLiteAVSDKTRTCMacx.x.x/SDK/TXLiteAVSDKTRTC_Mac.framework を、QTTest プロジェクトディレクトリと
 同じクラスのディレクトリにあるSDKフォルダにコピーします。

手順2:QTTest.proの設定

QTTest プロジェクトディレクトリを開き、任意のテキストエディタを使用して QTTest.pro ファイルを開いて から、SDK関連の引用を追加します。

```
INCLUDEPATH += $$PWD/.
DEPENDPATH += $$PWD/.
LIBS += "-F$$PWD/base/util/mac/usersig"
LIBS += "-F$$PWD/../SDK"
LIBS += -framework TXLiteAVSDK_TRTC_Mac
LIBS += -framework Accelerate
LIBS += -framework AudioUnit
INCLUDEPATH += $$PWD/../SDK/TXLiteAVSDK_TRTC_Mac.framework/Headers/cpp_interface
INCLUDEPATH += $$PWD/base/util/mac/usersig/include
DEPENDPATH += $$PWD/base/util/mac/usersig/include
```

手順3:カメラおよびマイクの使用権限の付与

SDKではカメラおよびマイクを使用しますので、対応する Info.plist に該当する権限申請説明を追加する必要 があります。

NSMicrophoneUsageDescription:マイクの使用申請 NSCameraUsageDescription:カメラの使用申請

下図に示すとおり:

手順4:TRTC SDKの引用

1.ヘッダーファイル #include "ITRTCCloud.h" によって直接引用することができます。
 2. ネームスペースの利用: C++ のすべてのプラットフォームのインターフェースのメソッド、タイプなどはいずれも trtc ネームスペースで定義されています。コードをより簡潔にするため、trtc ネームスペースを直接使用することをお勧めします。

説明:

ここまでで統合作業はすでに完了していますので、プロジェクトをコンパイルして実行できます。 Demo を使用するためのクロスプラットフォームSDKのAPIの詳細については、QTDemo をダウンロードして詳 細をご参照ください。

Windows端末の統合

開発環境要件

- OS: Windows 7以上のバージョン。
- 開発環境: Visual Studio 2015以上のバージョン、Visual Studio 2015の使用をお薦めします。VS関連のQT 開発環境をすでに設定していることが前提になります。

説明:

VS 関連のQT開発環境の設定手順に詳しくない場合は、 README の第2条の内容をご参照ください。

操作手順

ここでは、簡単なQTTestプロジェクトの新規作成を例に、 Visual StudioプロジェクトでC++クロスプラットフォームSDKを統合する方法をご紹介します。

手順1: C++クロスプラットフォームSDKのダウンロード

- 1. SDKをダウンロードし、ファイルを解凍して開きます。
- 2. QTTestと同じクラスのディレクトリ下で空のSDKフォルダを作成し、手順1でダウンロードし
- た TXLiteAVSDKTRTCWin_latest/SDK/CPlusPlus を、QTTest プロジェクトディレクトリと同じクラスのディレ クトリにあるSDKフォルダにコピーします。

手順2:QTTestプロジェクトリクエスト環境の設定

シーン1:QtCreatorを使用したリクエスト環境の設定

QTTestプロジェクトディレクトリを開き、任意のテキストエディタ(Sublime Textを推奨)を使用して QTTest.pro (Qt Creatorを使用して作成)ファイルを開いてから、SDKに関連する引用を追加します。

```
INCLUDEPATH += $$PWD/.
$$PWD/../SDK/CPlusPlus/Win32/include ¥
$$PWD/../SDK/CPlusPlus/Win32/include/TRTC
DEPENDPATH += $$PWD/.
$$PWD/../SDK/CPlusPlus/Win32/include ¥
$$PWD/../SDK/CPlusPlus/Win32/include/TRTC
CONFIG += opengl
CONFIG += debug and release
debug {
contains(QT ARCH, i386) {
LIBS += -L$$PWD/../SDK/CPlusPlus/Win32/lib -lliteav
} else {
LIBS += -L$$PWD/../SDK/CPlusPlus/Win64/lib -lliteav
}
}
release {
contains(QT_ARCH, i386) {
LIBS += -L$$PWD/../SDK/CPlusPlus/Win32/lib -lliteav
} else {
LIBS += -L$$PWD/../SDK/CPlusPlus/Win64/lib -lliteav
}
}
```

シーン2:VSを使用したリクエスト環境の設定

プロジェクトがすでに本格的なVSプロジェクトになっている場合は、VSのプロジェクト属性 Properties->Linker->Input およびGeneral でSDKライブラリパスのリクエスト情報を設定し、同時に Properties -> C/C++ -> General でSDK のヘッダーファイルパスのリクエスト情報を設定します。

手順3:ファイルのコピー

VSを使用して QTTest.pro プロジェクトを開き、関連する debug/release フォルダを自動的に生成してか ら、 SDK/CPlusPlus/Win32/lib 下のすべての .dll ファイルをプロジェクトディレクトリ下の debug/release フォルダにそれぞれコピーする必要があります。

手順4:TRTC SDKの引用

1.ヘッダーファイル #include "ITRTCCloud.h" によって直接引用することができます。
 2. ネームスペースの利用: C++ のすべてのプラットフォームのインターフェースのメソッド、タイプなどはいずれも trtc ネームスペースで定義されています。コードをより簡潔にするため、trtc ネームスペースを直接使用することをお勧めします。

説明:

ここまでで統合作業はすでに完了していますので、プロジェクトをコンパイルして実行できます。 Demo を使用するためのクロスプラットフォームSDKのAPIの詳細については、QTDemo をダウンロードして詳 細をご参照ください。

クイックインテグレーション(Unity)

最終更新日::2021-09-29 15:49:53

ここでは、主にTencent Cloud TRTC SDK(Unity)をプロジェクトに素早く統合する方法をご紹介します。以下の手順に従って設定すれば、SDKの統合作業を完了できます。

環境要件

- Unityの推奨バージョン: 2020.2.1f1c1。
- 現在、Android、iOS、Windows、Mac (Macはベータ版テスト中です)プラットフォームをサポートしています。
- Android Build Support 、 iOS Build Support 、 Winodows Build Support および MacOs Build Support モ ジュールが含まれている必要があります。
- その内、iOS端末の開発には以下が必要です。
 - 。 Xcode 11.0およびそれ以降のバージョン。
 - 。 プロジェクトが有効な開発者による署名を設定済みであることを確認してください。

SDKの統合

- 1. SDKおよび付属するDemoソースコードをダウンロードします。
- 2. 解凍後、プロジェクト内の TRTCUnitySDK/Assets/TRTCSDK/SDK フォルダを自分のプロジェクトのAssetsディ レクトリ下にコピーします。

よくあるご質問

Androidではネットワーク権限の問題が表示されますか。

プロジェクト内の /Assets/Plugins/AndroidManifest.xml ファイルを同じレベルのディレクトリ下に配置してく ださい。

Androidではオーディオビデオの権限はないのですか。

Android端末のマイク、カメラの権限は手動で申請します。具体的な方法については以下のコードをご参照ください。

#if PLATFORM_ANDROID

if (!Permission.HasUserAuthorizedPermission(Permission.Microphone))

{
Permission.RequestUserPermission(Permission.Microphone);
}
if (!Permission.HasUserAuthorizedPermission(Permission.Camera))
{
Permission.RequestUserPermission(Permission.Camera);
}
#endif

クイックインテグレーション(React Native)

最終更新日:::2021-12-31 15:29:36

ここでは、主にTencent Cloud TRTC SDK(React Native)をプロジェクトに素早く統合する方法をご紹介しま す。以下の手順に従って設定すれば、SDKの統合作業を完了できます。

環境要件

- ReactNative 0.63およびそれ以降のバージョン。
- Node & Watchman。nodeバージョンはv12以上が必要です。
- Android端末向け開発:
 - 。 Android Studio 3.5およびそれ以降のバージョン。
 - 。 AppにはAndroid 4.1およびそれ以降のバージョンのデバイスが必要です。
 - Java Development Kit
- iOS & macOS端末向け開発:
 - 。 Xcode 11.0およびそれ以降のバージョン。
 - 。 osxシステムには10.11およびそれ以降のバージョンが必要です。
 - 。 プロジェクトが有効な開発者による署名を設定済みであることを確認してください。
- 環境構築については、公式ドキュメントをご参照ください。

SDKの統合

ReactNative SDKは、npmですでに公開されています。 package. json を設定することでセットアップできます。

1. プロジェクトの package.json に以下の依存を書き入れます。

```
"dependencies": {
  "trtc-react-native": "^2.0.0"
},
```

- 2. カメラとマイクの許可が有効になると、音声通話機能が起動します。
 - 。 Android端末
 - 。 iOS端末
 - i. AndroidManifest.xml の中でAppの権限を設定します。TRTC SDKでは以下の権限が必要です。

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera" />
</uses-feature android:name="android.hardware.camera" />
</uses-feature android:name="android.hardware.camera" />
</uses-feature android:name="android.hardware.camera" />
</uses-feature android:name="android

注意

android:hardwareAccelerated="false" は設定しないでください。ハードウェアアクセラレーション を無効にすると、相手側のビデオストリームがレンダリングできなくなります。

ii. Android端末のオーディオ/ビデオの権限は手動でリクエストする必要があります。

```
if (Platform.OS === 'android') {
await PermissionsAndroid.requestMultiple([
PermissionsAndroid.PERMISSIONS.RECORD_AUDIO, //オーディオが必要
PermissionsAndroid.PERMISSIONS.CAMERA, //ビデオが必要
]);
}
```

クイックインテグレーション(Unreal Engine)

最終更新日:::2022-04-02 13:19:22

ここでは、主にTencent Cloud TRTC SDK(Unreal Engine)をプロジェクトに素早く統合する方法をご紹介しま す。以下の手順に従って設定すれば、SDKの統合作業を完了できます。

環境要件

- Unreal Engine 4.27.1およびそれ以降のバージョンを推奨します。
- Android端末向け開発:
 - 。 Android Studio 4.0およびそれ以降のバージョン。
 - 。 Visual Studio 2017 15.6以上のバージョン。
 - 。 実機デバッグのみサポートしています

iOS & macOS端末向け開発:

- Xcode 11.0およびそれ以降のバージョン。
- osxシステムには10.11およびそれ以降のバージョンが必要です。
- プロジェクトが有効な開発者による署名を設定済みであることを確認してください。
- Windows 端末向け開発:
 - 。 OS:Windows 7 SP1およびそれ以降のバージョン(x86-64に基づく64ビットOS)。
 - ディスク容量:IDEと一部のツールのインストールに必要な容量を除く、少なくとも1.64 GB以上の空き容量を確保するようにしてください。
 - 。 Visual Studio 2019をインストールします。

SDKの統合

- 1. SDKおよび付属するSDKソースコードをダウンロードします(ご質問があればQQグループ番号:764231117 に参加してお問い合わせください)。
- 2. 解凍後、プロジェクト内の TRTCSDK フォルダを自分のプロジェクトのSource/[project_name]**ディレク トリ下にコピーします。この中の[project_name]**にはプロジェクトの名称が入ります。
- 3. プロジェクトの**[project_name].Build.cs**ファイルを編集し、以下の関数を追加します

```
// 各プラットフォームのTRTC下層データベースをロード
private void loadTRTCSDK(ReadOnlyTargetRules Target)
{
string _TRTCSDKPath = Path.GetFullPath(Path.Combine(ModuleDirectory, "TRTCSDK"));
bEnableUndefinedIdentifierWarnings = false;
if (Target.Platform == UnrealTargetPlatform.Android)
{
// Androidヘッダーファイルをロード
PublicIncludePaths. Add(Path. Combine( TRTCSDKPath, "include/Android"));
PrivateDependencyModuleNames.AddRange(new string[] { "Launch" });
// Android APLファイルをロード
AdditionalPropertiesForReceipt. Add(new ReceiptProperty("AndroidPlugin", Path.Combine(ModuleDir
ectory, "TRTCSDK", "Android", "APL_armv7.xml")));
string Architecture = "armeabi-v7a";
// string Architecture = "arm64-v8a";
// string Architecture = "armeabi";
PublicAdditionalLibraries. Add(Path.Combine(ModuleDirectory, "TRTCSDK", "Android", Architecture,
"libtraeimp-rtmp.so"));
PublicAdditionalLibraries. Add(Path.Combine(ModuleDirectory, "TRTCSDK", "Android", Architecture,
"libliteavsdk.so"));
}else if (Target.Platform == UnrealTargetPlatform.IOS)
{
// iOSヘッダーファイルをロード
PublicIncludePaths.Add(Path.Combine(_TRTCSDKPath, "include/iOS"));
PublicAdditionalLibraries.AddRange(new string[] {
"resolv",
"z",
"c++"
});
PublicFrameworks.AddRange(
new string[] {
"CoreML",
"VideoToolbox",
"Accelerate",
"CFNetwork",
"OpenGLES",
"AVFoundation"
"CoreTelephony"
}
);
PublicAdditionalFrameworks. Add(new UEBuildFramework( "TXLiteAVSDK TRTC", TRTCSDKPath+"/ios/TXL
iteAVSDK TRTC.framework.zip", ""));
}else if(Target.Platform == UnrealTargetPlatform.Mac)
{
// MacOsヘッダーファイルをロード
PublicIncludePaths.Add(Path.Combine(_TRTCSDKPath, "include/Mac"));
```

```
PublicAdditionalLibraries.AddRange(new string[] {
"resolv",
"z",
"c++"
"bz2",
});
PublicFrameworks.AddRange(
new string[] {
"AppKit",
"IOKit",
"CoreVideo",
"CFNetwork",
"OpenGl",
"CoreGraphics",
"Accelerate",
"CoreFoundation",
"SystemConfiguration",
"AudioToolbox",
"VideoToolbox",
"CoreTelephony",
"CoreWLAN",
"AVFoundation",
"CoreMedia",
"CoreAudio",
"AudioUnit"
"Accelerate",
});
PublicFrameworks. Add(Path.Combine(_TRTCSDKPath, "Mac", "Release", "TXLiteAVSDK_TRTC_Mac.framewo
rk"));
}else if (Target.Platform == UnrealTargetPlatform.Win64)
{
// Win64ヘッダーファイルをロード
PublicIncludePaths.Add(Path.Combine( TRTCSDKPath, "include/win64"));
PublicAdditionalLibraries. Add(Path.Combine(_TRTCSDKPath, "win64", "Release", "liteav.lib"));
PublicDelayLoadDLLs.Add(Path.Combine(_TRTCSDKPath, "win64", "Release", "liteav.dll"));
PublicDelayLoadDLLs. Add(Path.Combine(_TRTCSDKPath, "win64", "Release", "LiteAvAudioHook.dl
l"));
PublicDelayLoadDLLs. Add(Path.Combine(_TRTCSDKPath, "win64", "Release", "LiteAvAudioHookServic
e.dll"));
PublicDelayLoadDLLs. Add(Path.Combine(_TRTCSDKPath, "win64", "Release", "openh264.dll"));
PublicDelayLoadDLLs. Add(Path.Combine(_TRTCSDKPath, "win64", "Release", "TRAE.dll"));
RuntimeDependencies. Add("$(BinaryOutputDir)/liteav.dll", Path.Combine(_TRTCSDKPath, "win64",
"Release", "liteav.dll"));
RuntimeDependencies. Add("$(BinaryOutputDir)/LiteAvAudioHook.dll", Path.Combine( TRTCSDKPath,
"win64", "Release", "LiteAvAudioHook.dll"));
RuntimeDependencies. Add("$(BinaryOutputDir)/LiteAvAudioHookService.dll", Path.Combine( TRTCSDK
Path, "win64", "Release", "LiteAvAudioHookService.dll"));
RuntimeDependencies. Add("$(BinaryOutputDir)/openh264.dll", Path.Combine( TRTCSDKPath, "win64",
```

"Release", "openh264.dll")); RuntimeDependencies.Add("\$(BinaryOutputDir)/TRAE.dll", Path.Combine(_TRTCSDKPath, "win64", "Re lease", "TRAE.dll")); }

4. [project_name].Build.csファイルでこの関数を呼び出します

	TRTC_Demo.Build.cs — TRTC-Unreal-SDK
EXPLORER ····	C TRTC_Demo.Build.cs M ×
ッ TRTC-UNREAL-SDK 「日 日 ひ 日	TRTC_Demo > Source > TRTC_Demo > 😋 TRTC_Demo.Build.cs > 😭 TRTC_Demo > f
> Plugins	8 > private void loadDebugDll(ReadOnlyTargetRules Target)
> Saved	33
✓ Script	34 // TRTC SDK framework
✓ Source	1 reference
✓ TRTC_Demo	35 > private void loadTRICSDK(ReadUnlyTargetRules Target)
> debug	111 public TRTC Demo(ReadOnlyTargetRules Target) : base(Target)
> Private	112 {
	113 PCHUsage = PCHUsageMode.UseExplicitOrSharedPCHs;
> Android	114 PublicDependencyModuleNames.AddRange(new string[] { "Co
> include	115 // /, 116 // Uncommont if you are using Slate UT
> iOS	117 PrivateDependencyModuleNames_AddRange(new_string[] {"III
> Mac	118 //
> win64	119 loadTRTCSDK(Target);
C TRTC Domo Build oc M	120 loadDebugDll(Target);
	121 // Uncomment if you are using online features
Gen TRTC_Demo.cpp	122 // PrivateDependencyModuleNames.Add("OnlineSubsystem");
C TRTC_Demo.h	123
G TRTC_DemoGameModeBase.cpp	124 // To include OnlineSubsystemSteam, add it to the plugi
C TRTC_DemoGameModeBase.h	125 }
TRTC_Demo.Target.cs	127 }
TRTC_DemoEditor.Target.cs	128

5. これでTRTC SDKの統合は完了し、cppファイルでTRTCを使用できるようになりました。 #include "ITRTCCloud.h"

```
// TRTCシングルトンオブジェクトを取得
#if PLATFORM_ANDROID
if (JNIEnv* Env = FAndroidApplication::GetJavaEnv()) {
void* activity = (void*) FAndroidApplication::GetGameActivityThis();
// Androidの場合はここで現在のコンテキストオブジェクトを渡す必要があります
pTRTCCloud = getTRTCShareInstance(activity);
}
#else
pTRTCCloud = getTRTCShareInstance();
```



```
#endif
// イベントコールバックを登録
pTRTCCloud->addCallback(this);
// バージョン番号を取得
std::string version = pTRTCCloud->getSDKVersion();
// 入室
trtc::TRTCParams params;
params.userId = "123";
params.roomId = 110;
params.roomId = 110;
params.userSig = GenerateTestUserSig().genTestUserSig(params.userId, SDKAppID, SECRETKEY);
pTRTCCloud->enterRoom(params, trtc::TRTCAppSceneVideoCall);
```

パッケージ化

- macOS 端末
- Windows 端末
- iOS 端末
- Android 端末
- 1. File -> Package Project -> Mac
- 2. 権限を設定します。前の手順でコンパイルしたxxx.appファイルを右クリックし、「パッケージの内容を表示」を選択します

MacNoEditor			· 🖞 🟈	··· ~	Q
.git >	Binaries		TC_Demo.app		
gitignore	🛅 Build	>			
i .vscode >	🚞 Config	>			
ther-dll-source	🚞 Content	>			
📄 README.md	🚞 DerivedDataCache	>			
TRTC_Demo >	🚞 Intermediate	>			
	ios	>			
	MacNoEditor	>			
	Plugins	>			
	🛄 README.md				TRTC_Demo.app
	Saved	>			- 359.2 MB
	Cript Script	>			100
	Source	>			
	TRTC_Demo.uproject				
	TRTC_Demo.xcworkspace	1			11

3. 「Contents->Info.plist」に進みます

4. 「Information Property List」を選択し、次の2つの権限を追加します。

<key>NSCameraUsageDescription</key> <string>通常のビデオ通話が行えるようにカメラを許可します</string> <key>NSMicrophoneUsageDescription</key> <string>通常の音声通話が行えるようにマイクの権限を承認します</string>

5. UE4のeditorで実行中の場合は、**UE4Editor.app**ファイルを見つけて、上記の手順に従って権限を追加する 必要があります。

TRTC全プラットフォーム C++ APIドキュメント

中国語ドキュメント

英語ドキュメント

Demo クイックスタート

Demoクイックスタート(iOS&Mac)

最終更新日:::2022-03-30 16:15:47

ここでは、主にTRTC-API-Example(iOS&Mac)を素早く実行する方法をご紹介します。

環境要件

- Xcode 11.0およびそれ以降のバージョン。
- プロジェクトが有効な開発者による署名を設定済みであることを確認してください。
- Qt Creator 4.13.3(Mac)およびそれ以降のバージョン。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

ステップ1:新規アプリケーションの作成

 TRTCコンソールにログインし、【開発支援】>【Demoのクイック実行】を選択します。
 【アプリケーションの作成】をクリックして、 TestTRTC などのアプリケーション名を入力します。すでにア プリケーションがある場合は、【既存のアプリケーションを選択】をクリックします。

3. ビジネスのニーズに合わせてタグを追加または編集し、【作成する】をクリックします。

1 Create Ap	plication >	2 Download Source Code	>	3 Modify Configuration	>	4 Compile and Run
Application Type	O New C Exist	ing				
Application Name	TestTRTC					
Tag 🛈	Tags allow you to m + Add	anage resources by category. If existin	ng tags do n	ot meet your requirements, you	can manage ta	gs here 🗳 .
Create	Reset					

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15 文字以内とします。
- タグは、Tencent Cloudのさまざまなリソースを識別および整理するのに役立ちます。例えば:企業に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、TRTCアプリケーションにタグを追加することで部門情報にマークを付けることができます。タグは必須ではありません。実際のビジネスニーズに合わせてタグを追加または編集できます。

ステップ2:SDKおよびTRTC-API-Exampleのソースコードのダウンロード

実際のビジネスニーズに合わせてSDKおよびTRTC-API-Exampleのソースコードをダウンロードします。
 ダウンロード完了後、【ダウンロード済み。次へ】をクリックします。

ステップ3: TRTC-API-Example プログラムファイルの設定

- 1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づき、対応する開発環境を選択します。
- 2. LiteAVSDK_TRTC_iOS_バージョンナンバ/TRTC-API-Example/Debug/GenerateTestUserSig.h ファイルを見つけ て開きます。
- 3. GenerateTestUserSig.h のファイルの関連するパラメータを設定します。
 - 。 SDKAPPID: デフォルトは0、実際のSDKAppIDを設定してください。
 - 。 SECRETKEY: デフォルトは空文字列。実際のキー情報を設定してください。
- 4. 貼り付け完了後、【貼り付けました。次へ】をクリックすれば、作成が完了します。
- 5. コンパイル完了後、【コンソールの概要ページに戻る】をクリックすればOKです。

注意:

- ここで言及したUserSigの作成法は、クライアントコードにSECRETKEYを設定しますが、この手法の SECRETKEYは逆コンパイルによって逆クラッキングされやすく、キーがいったん漏洩すると、攻撃者 はTencent Cloudトラフィックを盗用できるようになります。そのためこの手法は、ローカルのTRTC-API-Exampleクイックスタートおよび機能デバッグにのみ適しています。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。 UserSigが必要なときは、Appから業務サーバーにリクエストを発出しダイナミック

UserSigを取得します。詳細はサーバーでのUserSig新規作成をご参照ください。

ステップ4:コンパイルと実行

XCode(11.0およびそれ以降のバージョン)を使用してソースディレクトリ配下の TRTC-API-Example-OC.xcworkspace プロジェクトを開き、 TRTC-API-Exampleプロジェクトをコンパイルして実行します。

よくあるご質問

キーをクエリーするとき、パブリックキーとプライベートキーの情報しか取得できませんが、キーはどうしたら取得できますか。

TRTC SDK 6.6バージョン(2019年08月)では、新しい署名アルゴリズムのHMAC-SHA256の使用を開始しました。それ以前に作成されたアプリケーションの場合、新しい暗号化鍵を取得するために、署名アルゴリズムをアップグレードする必要があります。アップグレードしない場合でも、旧バージョンアルゴリズム ECDSA-SHA256は引き続き使用できます。アップグレード済みなら、必要に応じて新旧アルゴリズムを切り替えます。

アップグレード/切替の操作:

1. Tencent Real-Time Communicationコンソールにログインします。

 2. 左側ナビゲーションバーで【アプリケーション管理】を選択し、ターゲットアプリケーションのある行の【ア プリケーション情報】をクリックします。

3. 【クイックスタート】タブを選択して【ステップ2 UserSigを発行するためのキーを取得】エリアの【ここ をクリックしてアップグレード】、【非対称暗号化】または【HMAC-SHA256】をクリックします。

- アップグレード:
- 旧バージョンアルゴリズムのECDSA-SHA256に切り替えます。

Step 2: obtain the secret key to issue UserSig				
The secret key is sensitive information. Please do not disclose it.				
Secret Key (Key)				
3d3f69f8fa29e161531ad44df1a1739d6358202539c7574c6 96e30e3caad4437				
Copy Secret Key				
* The current mode is "HMAC-SHA256", you can switch to "asymmetric encryption".				

• 新バージョンアルゴリズムのHMAC-SHA256に切り替えます。



2.2台の携帯電話で同時にプロジェクトを実行しているのに、お互いの画面が表示されないのはな ぜですか。

2台の携帯電話でプロジェクトを実行するとき、UserIDが異なるものを使用してください。TRTCでは、同一の UserID(SDKAppIDが異なる場合を除く)が2つの端末で同時に使用することをサポートしていません。

3. ファイアウォールにはどのような制限がありますか。

SDK が UDP プロトコルを使用してオーディオビデオ伝送を行っていることから、 UDPに対してブロックがある オフィスネットワークでは使用することができません。類似した問題がおありの際は、 企業ファイアウォール制 限の対応をご参照の上、問題及び原因解決にお役立てください。

Android

最終更新日:::2022-03-30 16:20:00

このドキュメントでは、主にTRTC-API-Example(Android)をすばやく実行する方法について説明します。

環境要件

- 互換性のある最低バージョンはAndroid 4.1 (SDK API Level 16)。Android 5.0 (SDK API Level 21)およびそれ以降のバージョンの使用を推奨します。
- Android Studio 3.5およびそれ以降のバージョン。
- AppにはAndroid4.1およびそれ以降のデバイスが必要です。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

ステップ1:新規アプリケーションの作成

- 1. TRTCコンソールにログインし、【開発支援】>【快速跑通Demo】を選択します。
- 【アプリケーションの作成】をクリックして、 TestTRTC などのアプリケーション名を入力します。すでにア プリケーションがある場合は、【既存のアプリケーションを選択】をクリックします。
- 3. ビジネスのニーズに合わせてタグを追加または編集し、【作成する】をクリックします。

1 Create Ap	plication > 2 Download Source Code	> (3) Modify Configuration	> (4) Compile and Run
Application Type	O New C Existing		
Application Name	TestTRTC		
Tag 🕄	Tags allow you to manage resources by category. If exis + Add	sting tags do not meet your requirements, you ca	n manage tags here .
Create	Reset		

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15 文字以内とします。
- タグは、Tencent Cloudのさまざまなリソースを識別および整理するのに役立ちます。例えば:企業 に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、TRTCアプリケー ションにタグを追加することで部門情報にマークを付けることができます。タグは必須ではありません。実際のビジネスニーズに合わせてタグを追加または編集できます。

ステップ2:SDKおよびTRTC-API-Exampleのソースコードのダウンロード

実際のビジネスニーズに合わせてSDKおよびTRTC-API-Exampleのソースコードをダウンロードします。
 ダウンロード完了後、【ダウンロード済み。次へ】をクリックします。

ステップ3:TRTC-API-Exampleプログラムファイルの設定

- 1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づき、対応する開発環境を選択します。
- LiteAVSDK_TRTC_Androidバージョンナンバ/TRTC-API Example/Debug/src/main/java/com/tencent/trtc/debug/GenerateTestUserSig.java ファイルを見つけて開き ます。
- 3. GenerateTestUserSig.java ファイル内の関連パラメータを設定します。
 - 。 SDKAPPID: デフォルトはPLACEHOLDER、実際のSDKAppIDを設定してください。
 - 。 SECRETKEY: デフォルトはPLACEHOLDER、実際のキー情報を設定してください。
- 4. 貼り付け完了後、【貼り付けました。次へ】をクリックすれば、作成が完了します。
- 5. コンパイル完了後、【コンソールの概要ページに戻る】をクリックすればOKです。

注意:

- ここで言及したUserSigの作成法は、クライアントコードにSECRETKEYを設定しますが、この手法の SECRETKEYは逆コンパイルによって逆クラッキングされやすく、キーがいったん漏洩すると、攻撃者 はTencent Cloudトラフィックを盗用できるようになります。そのためこの手法は、ローカルのAppク イックスタートおよび機能デバッグにのみ適しています。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appから業務サーバーにリクエストを発出し動的にUserSig を取得します。詳細はサーバーでのUserSig新規作成をご参照ください。

手順4:コンパイル動作

Android Studio(3.5およびそれ以降のバージョン)を使用してソースプロジェクト TRTC-API-Example を開 き、【実行】をクリックすればOKです。

よくあるご質問

キーをクエリーするとき、パブリックキーとプライベートキーの情報しか取得できませんが、キーはどうしたら取得できますか。

TRTC SDK 6.6バージョン(2019年08月)では新しい署名アルゴリズムのHMAC-SHA256の使用を始めていま す。その前に作成済のアプリケーションは、署名アルゴリズムをアップグレードしないと暗号化したキーを取得 できません。アップグレードしなくても、旧バージョンアルゴリズム ECDSA-SHA256は引き続き使用できま す。アップグレード済みなら、必要に応じて新旧アルゴリズムに切り替えます。

アップグレード/切替の操作:

1. Tencent Real-Time Communicationコンソールにログインします。

 左側ナビゲーションバーで【アプリケーション管理】を選択し、ターゲットアプリケーションのある行の【ア プリケーション情報】をクリックします。

3. 【クイックスタート】タブを選択して【ステップ2 UserSigを発行するためのキーを取得】エリアの【ここ をクリックしてアップグレード】、【非対称暗号化】または【HMAC-SHA256】をクリックします。

- アップグレード。
- 旧バージョンアルゴリズムの ECDSA-SHA256に切り替えます。



• 新バージョンアルゴリズムのHMAC-SHA256に切り替えます。

Step 2: obtain the secret key to issue UserSig				
The secret key is sensitive information. Please	e do not disclose it.			
Public Key (PublicKey)	Private Key (PrivateKey):			
- H S E				
	IXgEND PRIVATE KEY			
Copy Public Key	Copy Private Key			
* The current mode is "asymmetric encryption", you can switch to HMAC-SHA256".				

2.2台の携帯電話で同時にAppを実行しているのに、お互いの画面が表示されないのはなぜですか。

2台の携帯電話でAppを操作するとき、UserIDが異なるものを使用してください。TRTCでは、同一のUserID (SDKAppIDが異なる場合を除く)が2つの端末で同時に使用することをサポートしていません。

3. ファイアウォールにはどのような制限がありますか。

SDK が UDP プロトコルを使用してオーディオビデオ伝送を行っていることから、 UDPに対してブロックがある オフィスネットワークでは使用することができません。類似した問題がおありの際は、 企業ファイアウォール制 限の対応をご参照の上、問題及び原因解決にお役立てください。

Demoクイックスタート(Windows)

最終更新日::2022-03-30 16:24:04

ここでは、主にTRTC Demo(Windows)を素早く実行する方法をご紹介します。

環境要件

Windows(C++)開発環境

- Microsoft Visual Studio 2015バージョン以上、Microsoft Visual Studio 2015のご使用を推奨します。
- Windows SDK バージョン8.0以上、Windows SDK 8.1の使用を推奨します。

Windows(C#)開発環境

- Microsoft Visual Studio 2015バージョン以上、Microsoft Visual Studio 2017の使用を推奨します。
- .Net Framework バージョン4.0以上、.Net Framework 4.0の使用を推奨します。

Windows(QT)開発環境

- Microsoft Visual Studio 2015およびそれ以降のバージョン。Microsoft Visual Studio 2015の使用を推奨し ます。
- .vsix プラグインファイルをダウンロードしてインストールします。公式サイトで対応するプラグインのバージョンをさがしインストールすればOKです。
- VSを開いてツールバーから QT VS Tools -> Qt Options -> Qt Versions をさがし、自分のQtコンパイラ msvcをadd(追加)します。
- SDK/CPlusPlus/Win32/lib 下のすべての .dll ファイルをプロジェクトディレクトリ下の debug / release フォルダ下にコピーする必要があります。

注意:

debug/release フォルダは、すべて VS 上の環境設定が完了後に自動的に生成されます。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

ステップ1:新規アプリケーションの作成

1. TRTCコンソールにログインし、【開発支援】>【Demoのクイック実行】を選択します。

2.【アプリケーションの作成】をクリックして、 TestTRTC などのアプリケーション名を入力します。すでにア プリケーションがある場合は、【既存のアプリケーションを選択】をクリックします。

3. ビジネスのニーズに合わせてタグを追加または編集し、【作成する】をクリックします。

1 Create Ap	plication > 2 Download Source Code	> (3) Modify Configuration	> (4) Compile and Run
Application Type	• New Existing		
Application Name	TestTRTC		
Tag 🚯	Tags allow you to manage resources by category. If existi + Add	ing tags do not meet your requirements, you ca	in manage tags here .
Create	Reset		

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15文 字以内とします。
- タグは、Tencent Cloudのさまざまなリソースを識別および整理するのに役立ちます。例えば:企業に 複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、TRTCアプリケーション にタグを追加することで部門情報にマークを付けることができます。タグは必須ではありません。実際 のビジネスニーズに合わせてタグを追加または編集できます。

ステップ2:SDKおよびDemoソースコードをダウンロード

1. 実際のビジネスニーズに基づき、SDKおよび付属のDemoソースコードをダウンロードします。
 2. ダウンロード完了後、【ダウンロード済み。次へ】をクリックします。

ステップ3: Demo プログラムファイルの設定

1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づき、対応する開発環境を選択します。

2. GenerateTestUserSig ファイルを見つけて開きます。:
| 適用可能なプラットフォーム | ファイル相対パス |
|---------------|---|
| Windows(C++) | Windows/DuilibDemo/GenerateTestUserSig.h |
| Windows(C#) | Windows/CSharpDemo/GenerateTestUserSig.cs |

3. GenerateTestUserSig.js のファイルの関連するパラメータを設定します。

- 。SDKAPPID:デフォルトは0。実際のSDKAppIDを設定してください。
- 。 SECRETKEY: デフォルトは空文字列。実際のキー情報を設定してください。
- 4. 貼り付け完了後、【貼り付けました。次へ】をクリックすれば、作成が完了します。
- 5. コンパイル完了後、【コンソールの概要ページに戻る】をクリックすればOKです。

注意:

- ここで言及したUserSigの新規作成ソリューションでは、クライアントコードでSECRETKEYを設定します。この手法のうちSECRETKEYは逆コンパイルによって逆向きにクラッキングされやすく、キーがいったん漏洩すると、攻撃者はTencent Cloudトラフィックを盗用できるようになります。そのためこの 手法は、ローカルのDemoクイックスタートおよび機能デバッグにのみ適合します。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appから業務サーバーにリクエストを発出し動的にUserSig を取得します。詳細はサーバーでのUserSig新規作成をご参照ください。

ステップ4:コンパイル実行

• Windows(C++):

Visual Studio(VS2015をお勧めします)を使用してソースコードディレクトリ

の DuilibDemo¥TRTCDuilibDemo.sln プロジェクトファイルを開きます。Release/X86 を選択してプラットフォームを構築することをお勧めします。Demoプロジェクトをコンパイルし動作させればOKです。

Windows(C#):

Visual Studio (VS2017をお勧めします)を使用してソースコードディレクトリ

の CSharpDemo¥TRTCCSharpDemo.sln プロジェクトファイルを開きます。Release/X86を選択してプラットフォームを構築することをお勧めします。Demoプロジェクトをコンパイルし動作させればOKです。

Windows(QT):

Visual Studio(VS2015以上を推奨)を使用してソースコードディレクトリの QTDemo\QTDemo.proプロジェクトファイルを開きます。QTDemo プロジェクトをコンパイルし動作させれ ばOKです。

よくあるご質問

1. キーをクエリーするとき、公開鍵および秘密鍵の情報しか取得できませんが、キーはどうしたら 取得できますか。

TRTC SDK 6.6バージョン(2019年08月)では、新しい署名アルゴリズムのHMAC-SHA256の使用を開始しました。それ以前に作成されたアプリケーションの場合、新しい暗号化鍵を取得するために、署名アルゴリズムをア ップグレードする必要があります。アップグレードしない場合でも、旧バージョンアルゴリズム ECDSA-SHA256は引き続き使用できます。アップグレード済みなら、必要に応じて新旧アルゴリズムを切り替えます。

アップグレード/切替の操作:

- 1. Tencent Real-Time Communicationコンソールにログインします。
- 左側ナビゲーションバーで【アプリケーション管理】を選択し、ターゲットアプリケーションのある行の【ア プリケーション情報】をクリックします。

3. 【クイックスタート】タブを選択して【ステップ2 UserSigを発行するためのキーを取得】エリアの【ここ をクリックしてアップグレード】、【非対称暗号化】または【HMAC-SHA256】をクリックします。

- アップグレード:
- 旧バージョンアルゴリズムのECDSA-SHA256に切り替えます。

Step 2: obtain the secret key to issue UserSig
The secret key is sensitive information. Please do not disclose it.
Secret Key (Key)
3d3f69f8fa29e161531ad44df1a1739d6358202539c7574c6 96e30e3caad4437
Copy Secret Key
* The current mode is "HMAC-SHA256" you can switch to "asymmetric operation

• 新バージョンアルゴリズムのHMAC-SHA256に切り替えます。



2.2台のデバイスで同時に**Demo**を実行しているのに、お互いの画面が表示されないのはなぜですか?

2台のデバイスでDemoを実行するときは、異なるUserIDを使用していることを確認してください。TRTCでは、 2台のデバイスでの同一UserID(SDKAppIDが異なる場合を除く)の同時使用をサポートしていません。

3. ファイアウォールにはどのような制限がありますか。

SDKがUDPプロトコルを使用してオーディオ・ビデオ伝送を行っているため、UDPをブロックするオフィスネットワークでは使用できません。類似した問題がおありの際は、企業ファイアウォール制限の対応をご参照の上、 問題及び原因解決にお役立てください。

Web

最終更新日::2022-04-02 16:27:03

ここでは、主にTencent Cloud TRTC Web SDK Demoを素早く実行する方法をご紹介します。

準備作業

TRTC Web SDK Demoを実行する前に理解すべきの事項。

サポートするプラットフォーム

TRTC Web SDKはWebRTCに基づき実現され、現在、デスクトップとモバイル端末の主流ブラウザをサポートしています。詳細なサポートレベルの表については、サポートしているプラットフォームをご参照ください。 お客様のユースケースは対応表に記載されていない場合は、TRTC Web SDK機能テスト画面を開けて、WebView など、現在の環境がWebRTCのすべての機能をサポートしているかどうかをチェックすることができます。

お客様のユースケースが主に教育シーンである場合は、教師用端末ではElectronソリューションの使用をお勧めし、大小2チャネル画面、よりフレキシブルな画面共有方法およびより強力な弱ネットワークリカバリー機能をサポートしています。

URLドメイン名プロトコルの制限

ブラウザセキュリティポリシーの制限により、WebRTC機能を使用したページへのアクセスプロトコルには厳し い要件があります。以下の表を参照してアプリケーションの開発とデプロイを行ってください。

ユースケース	プロトコル	受信 (再生)	送信(マイク・ オン)	画面共有	備考
本番環境	HTTPSプロトコル	サポートあ り	サポートあり	サポートあ り	推奨
本番環境	HTTPプロトコル	サポートあ り	サポートなし	サポートな し	
ローカル開発環 境	http://localhost	サポートあ り	サポートあり	サポートあ り	**推 奨
ローカル開発環 境	http://127.0.0.1	サポートあ り	サポートあり	サポートあ り	
ローカル開発環 境	http://[ローカルマシ ンIP]	サポートあ り	サポートなし	サポートな し	

ユースケース	プロトコル	受信 (再生)	送信(マイク・ オン)	画面共有	備考
ローカル開発環 境	file:///	サポートあ り	サポートあり	サポートあ り	

ファイアウォールの制限

TRTC Web SDKは次のポートとドメイン名に依存してデータ転送を行い、ファイアウォールのホワイトリストに 追加してください。設定が完了したら、公式サイトDemoにアクセスし体験して設定が有効になっているかを確 認することができます。具体的には、ファイアウォール制限の対応関連をご参照ください。

- TCPポート:8687
- UDPポート:8000、8080、8800、843、443、16285
- ドメイン名: *.rtc.qq.com , yun.tim.qq.com

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

手順1:アプリケーションの新規作成

1. TRTCコンソールにログインし、開発支援 > **Demoクイックスタート**を選択します。**

2. アプリケーションの作成をクリックし、 TestTRTC などのアプリケーション名を入力します。すでにアプ

- **リケーションを作成している場合、**既存のアプリケーションを選択**をクリックします。
- 2. 実際のサービスニーズに応じてタグを追加または編集し、作成をクリックします。

1 Create Ap	plication >	2 Download Source Code	> 3 Modify Configuration	> (4) Compile and Run
Application Type	◯ New 🔵 Exist	ing		
Application Name	TestTRTC			
Tag	Tags allow you to m + Add	anage resources by category. If existi	ng tags do not meet your requirements, you ca	an manage tags here .
Create	Reset			

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15文 字以内とします。
- タグはTencent Cloudのさまざまなリソースを識別して管理するために使用されます。たとえば、企業 に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、企業はTRTCアプリケ ーションにタグを追加することで部門情報にマークを付けることができます。タグは入力必須ではあり ません。実際のサービスニーズに応じてタグを追加または編集できます。

手順2:SDKおよびDemoソースコードをダウンロード

1. 実際のサービスニーズに基づき、SDKおよび付属のDemoソースコードをダウンロードします。



O pownload SDK and Auxillary Demo Source Code Platform Operation Platform Operation DS Download at GitHub Download at Gite Download at GitHub Download at Gite Download Zip Veb Download at GitHub Download at Gite Download Zip AacOS Download at GitHub Download at Gite Download Zip Gitectron Download at GitHub Download at Gite Download Zip Vindows Download at GitHub Download at Gite Download Zip Gitter Download at GitHub Download at Gite Download Zip	Create Application	on > 2 Download Source Code	> (3) Modify Configuration	> (4) Compile and Run
iation Operation NS Download at GitHub Download at Gitee Download Zip Indroid Download at GitHub Download at Gitee Download Zip Ieb Download at GitHub Download at Gitee Download Zip IacOS Download at GitHub Download at Gitee Download Zip Iectron Download at GitHub Download at Gitee Download Zip Indows Download at GitHub Download at Gitee Download Zip Iuter Download at GitHub Download at Gitee Download Zip	Download SDK ar	nd Auxiliary Demo Source Code		
DS Download at GitHub Download at Gite Download Zp Indroid Download at GitHub Download at Gite Download Zp Veb Download at GitHub Download at Gite Download Zp NacOS Download at GitHub Download at Gite Download Zp Iectron Download at GitHub Download at Gite Download Zp Vindows Download at GitHub Download at Gite Download Zp Iutter Download at GitHub Download at Gite Download Zp	latform	Operation		
AndroidDownload at GitHubDownload at GiteeDownload ZipNebDownload at GitHubDownload at GiteeDownload ZipVacOSDownload at GitHubDownload at GiteeDownload ZipElectronDownload at GitHubDownload at GiteeDownload ZipNindowsDownload at GitHubDownload at GiteeDownload ZipFlutterDownload at GitHubDownload at GiteeDownload Zip	OS	Download at GitHub Download at Gite	ee Download Zip	
Web Download at GitHub Download at Gitee Download Zip MacOS Download at GitHub Download at Gitee Download Zip Electron Download at GitHub Download at Gitee Download Zip Mindows Download at GitHub Download at Gitee Download Zip Flutter Download at GitHub Download at GitHub Download at GitHub	ndroid	Download at GitHub Download at Gite	ee Download Zip	
MacOS Download at GitHub Download at Gitee Download Zip Electron Download at GitHub Download at Gitee Download Zip Mindows Download at GitHub Download at Gitee Download Zip Flutter Download at GitHub Download at GitHub Download at Gitee	Veb	Download at GitHub Download at Gite	ee Download Zip	
Electron Download at GitHub Download at Gitee Download Zip Mindows Download at GitHub Download at Gitee Download Zip Flutter Download at GitHub	lacOS	Download at GitHub Download at Gite	ee Download Zip	
Windows Download at GitHub Download at Gitee Download Zip Flutter Download at GitHub State State	lectron	Download at GitHub Download at Gite	ee Download Zip	
Flutter Download at GitHub	Vindows	Download at GitHub Download at Gite	ee Download Zip	
	lutter	Download at GitHub		
	Next Previous			

2. ダウンロード完了後、**ダウンロード完了、次のステップへ**をクリックします。

手順3:Demoプログラムファイルの設定

1.設定変更ページに進み、 SDKAppID と キー を取得します。

2.ダウンロードしたソースで Web/base-js/js/debug/GenerateTestUserSig.js'ファイルを探して開きます。 3. GenerateTestUserSig.js のファイルの関連するパラメータを設定します。 -SDKAPPID: デフォルトでは0で す。実際の SDKAppID を設定してください。 -SECRETKEY: デフォルトでは空文字列です。実際の キー`情報を

Create Application > Ownload Source > Code	3 Modify Configuration	>	4 Compile & Run				
Paste SDKAppID and Secret Key to	Decompress the sour bug/GenerateTestUse	te package downloa Sig.java File	ided in Step 2, and op	en Android/TRTCSce	nesDemo/d	ebug/src/main/java/o	com/tencent/liteav/de
Specified Location	Android	iOS&macOS	Windows(C++)	Windows(C#)	Web	Mini Program	Electron
SDKAppID Copy	public class Gener	ateTestUserSig {					
	private static	final int SDKAP	PID = 0;				
Secret Key Copy b	private static	final int EXPIRI	TIME = 604800;				
* The secret key is sensitive information. Please do not disclose it.							
		_					

4. 貼り付け完了後、コピーとペースト完了、次のステップへをクリックすると、作成が完了します。

注意:

Tencent Cloud

- ここで言及したUserSigの作成ソリューションでは、クライアントコードでSECRETKEYを設定します。
 この手法のうちSECRETKEYは逆コンパイルによって逆向きにクラッキングされやすく、キーがいったん漏洩すると、攻撃者はTencent Cloudトラフィックを盗用できるようになります。そのためこの手法は、ローカルのDemoクイックスタートおよび機能デバッグにのみ適合します。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appからサービスサーバーにリクエストを発出し動的に UserSigを取得します。詳細はサーバーでのUserSig作成をご参照ください。

手順4:Demoの実行

Chromeブラウザを使用してDemoルートディレクトリの index.html ファイルを開けば、Demoを実行できます。

注意:

- 通常の場合は、体験Demoは、サーバーにデプロイし、 https://ドメイン名/xxx 経由でアクセスする か、または直接ローカルにサーバーを構築して、 localhost:ポート経由で アクセスする必要がありま す。
- 現在デスクトップ版Chromeブラウザは、TRTC Web SDK関連機能のサポート状況がかなり整っていますので、Chromeブラウザを使用して体験することをお勧めします。
- ルーム追加をクリックして、オーディオビデオ通話ルームを追加し、ローカルのオーディオビデオストリーム をデプロイします。
 複数ページを開き、各画面でルーム追加をクリックすることができます。正常状態では、複数画面を見てリア ルタイムなオーディオビデオ通話をシミュレーションできます。
- カメラアイコンをクリックすると、カメラデバイスを選択できます。
- マイクのアイコンをクリックすると、マイクデバイスを選択できます。

説明:

WebRTCは、カメラとマイクを使用して、オーディオとビデオをキャプチャする必要があります。体験 中、Chromeブラウザから関連プロンプトが表示されることがありますが、その場合、**許可**をクリック します。

オンラインDemoの実行

また、Demoをオンラインで実行することもできます。 SDKAppID と キー を入手したら、Demoを素早く実行 するをクリックしてページの指示に従って実行してください。

よくあるご質問

キーをクエリーするとき、パブリックキーとプライベートキーの情報しか取得できませんが、キーはどうしたら取得できますか。

TRTC SDK 6.6 (Web SDK 4.0) バージョン(2019年08月)では、新しい署名アルゴリズムのHMAC-SHA256 の使用を開始しました。それ以前に作成されたアプリケーションの場合、新しい暗号化鍵を取得するために、署 名アルゴリズムをアップグレードする必要があります。アップグレードしない場合でも、旧バージョンアルゴリズム ECDSA-SHA256は引き続き使用できます。アップグレードされた場合は、必要に応じて新旧アルゴリズム を切り替えます。

アップグレード/切替の操作:

- 1. TRTCコンソールにログインします。
- 左側ナビゲーションバーでアプリケーション管理を選択し、ターゲットアプリケーションのある行のアプリケーションのある行のアプリケーション情報をクリックします。
- 3. クイックマスタータブを選択して手順2 UserSigを発行するためのキーを取得エリアのここをクリックしてア ップグレード、非対称暗号化またはHMAC-SHA256をクリックします。
- アップグレード:
- 旧バージョンアルゴリズムのECDSA-SHA256に切り替えます。

Step 2: obtain the secret key to issue UserSig
The secret key is sensitive information. Please do not disclose it.
Secret Key (Key)
3d3f69f8fa29e161531ad44df1a1739d6358202539c7574c6 96e30e3caad4437
Copy Secret Key * The current mode is "HMAC-SHA256", you can switch to "asymmetric encryption".

• 新バージョンアルゴリズムのHMAC-SHA256に切り替えます。



2. クライアントエラーの発生「RtcError: no valid ice candidate found」にはどう対処す ればよいでしょうか。

このエラーが発生した場合、TRTC Web SDKがSTUNトンネリングに失敗したことを意味しますので、環境要件 ファイアウォールの設定を確認してください。

3. クライアントエラーの発生:「RtcError: ICE/DTLS Transport connection failed"」ま たは「RtcError: DTLS Transport connection timeout」にはどう対処すればよいでしょう か。

このエラーが発生した場合、TRTC Web SDKがメディア伝送チャネルの確立に失敗したことを意味しますので、 環境要件ファイアウォールの設定を確認してください。

4.10006エラーが発生したときはどう対処すればよいでしょうか。

「Join room failed result: 10006 error: service is suspended,if charge is overdue,renew it」が発生した 場合は、TRTCアプリケーションのサーバー状態が正常かどうかをご確認ください。

TRTCコンソールにログインし、作成したアプリケーションをクリックし、アカウント情報をクリックすると、ア カウント情報画面でサービス状態を確認することができます。

TRTC Service Status

Status Available

説明:

その他のよくあるご質問については、Web端末に関するご質問をご参照ください。

Electron

最終更新日::2022-03-30 16:54:23

ここでは、主にTRTC Demo (Electron)を素早く実行する方法をご紹介します。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

手順1:アプリケーションの新規作成

- 1. TRTCコンソールにログインし、開発支援>Demoクイックスタートを選択します。
- アプリケーションの作成をクリックし、 TestTRTC などのアプリケーション名を入力します。すでにアプリケーションを作成している場合、既存のアプリケーションを選択をクリックします。
- 3. 実際の業務ニーズに応じてタグを追加または編集し、**作成**をクリックします。

1 Create App	Dication > 2 Download Code	Source >	3 Modify Configuration	> (4) Compile and Run
Application Type	O New Existing			
Application Name	TestTRTC			
Tag	Tags allow you to manage resources by cat + Add	egory. If existing tags do	o not meet your requirements, you	can manage tags here .
Create	Reset			

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15文字以内とします。
- タグはTencent Cloudのさまざまなリソースを識別して管理するために使用されます。たとえば、企業 に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、企業はTRTCアプリケ

ーションにタグを追加することで部門情報にマークを付けることができます。タグは入力必須ではあり ません。実際のビジネスニーズに応じてタグを追加または編集できます。

ステップ2:SDKおよびDemoソースコードをダウンロード

1.実際のビジネスニーズに基づき、SDKおよび付属のDemoソースコードをダウンロードします。
 2.ダウンロード完了後、ダウンロードしました。次のステップをクリックします。

Create Application	> 2 Download Source Code	> 3 Modify Config	> uration	4 Compile and Run
Download SDK and Auxiliar	ry Demo Source Code			
Platform	Operation			
iOS	Download at GitHub Download at Gitee	Download Zip		
Android	Download at GitHub Download at Gitee	Download Zip		
Web	Download at GitHub Download at Gitee	Download Zip		
MacOS	Download at GitHub Download at Gitee	Download Zip		
Electron	Download at GitHub Download at Gitee	Download Zip		
Windows	Download at GitHub Download at Gitee	Download Zip		
Flutter	Download at GitHub			
Next Previous				

ステップ3: Demo プログラムファイルの設定

1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づき、対応する開発環境を選択します。

- 2. Electron/js/GenerateTestUserSig.js ファイルを見つけて開きます。
- 3. GenerateTestUserSig.js のファイルの関連するパラメータを設定します。
 - 。 SDKAPPID:デフォルトは0。実際のSDKAppIDを設定してください。
 - 。 SECRETKEY: デフォルトは空文字列。実際のキー情報を設定してください。

Create Application > Create Application > Create Application >	3 Modify > (4) Compile & Run
Paste SDKAppID and Secret Key to	Decompress the source package downloaded in Step 2, and open Android/TRTCScenesDemo/debug/src/main/java/com/tencent/liteav/de bug/GenerateTestUserSig.java File
Specified Location	Android IOS&macOS Windows(C++) Windows(C#) Web Mini Program Electron
SDKAppID Copy Secret Key Copy	private static final int EXPIRETIME = 604800;
 The secret key is sensitive information. Please do not disclose it. 	
Pasted and Next	private static final String SECRETKEY = ``;

4. 貼り付け完了後、**貼り付けました。次のステップ**をクリックすれば、作成が完了します。

5. コンパイル完了後、 コンソール概要に戻る をクリックすれば終了です。

注意:

- ここで言及したUserSigの新規作成ソリューションでは、クライアントコードでSECRETKEYを設定します。この手法のうちSECRETKEYは逆コンパイルによって逆向きにクラッキングされやすく、キーがいったん漏洩すると、攻撃者はTencent Cloudトラフィックを盗用できるようになります。そのためこの 手法は、ローカルのDemoクイックスタートおよび機能デバッグにのみ適合します。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appから業務サーバーにリクエストを発出し動的にUserSigを 取得します。詳細はUserSigに関するご質問をご参照ください。

ファイルディレクトリの説明

|---README.md READMEファイルです。よくお読みください
|---main.electron.js Electronメインファイル
|---public 静的ファイルに保存
|---babel.config.js
|---package.json
|---vue.config.js vue-cliプロジェクトファイル
|---src ソースコードディレクトリ

| |---app.vue | |---common.css | |---main.js | |---components UIコンポーネントディレクトリ | | |---main-menu.vue | | |---nav-bar.vue | | ---show-screen-capture.vue | |---common ツール関数、パブリックコーパスなど | | |---live-room-service.js | | |---log.js ログツール | | |---mtah5.js | | |---routes.js | | |---rand.js | |---pages ページディレクトリ | | |---index.vue メインページ | | |---trtc ビデオミーティング関連ページ | | | |---trtc-room.vue ビデオミーティングルームページ | | | |---trtc-index.vue ビデオミーティングエントリーページ | | |---404.vue | | | ---live ライブストリーミングページ | | | | ---live-index.vue ライブストリーミングエントリーページ | | | | ---live-room-audience.vue ライブルーム視聴者ページ | | | |---live-room-anchor.vue ライブルームキャスターのページ | |---debug 注意!デプロイするとき、このフォルダ内の署名ロジックをサーバーに移動して実装してくだ さい | | |---lib-generate-test-usersig.min.js | | |---gen-test-user-sig.js

ステップ4:コンパイル実行

- Windowsプラットフォーム
- MacOSプラットフォーム
- 1. Nodeの最新バージョンをインストールします。64bitの .msi ファイルの使用を推奨します。Nodeダウンロ ードアドレス。
- win + r を押してcmdを入力し、管理者権限でコマンドラインウィンドウを起動し、プロジェクトディレクトリのディレクトリを見つけて、以下のコマンドを実行します。

\$ npm install



4. npmの依存パッケージのインストール完了を待って、コマンドラインウィンドウで以下のコマンドを引き続き 実行してDemoを実行します。

\$ npm run start # 初回の実行の場合、しばらくしてからウィンドウにUIが現れます



npm - C	
E:\Dev\web\trtc-electron-simple-demo>npm run start	<u>^</u>
> trtc-electron-simple-demo@0.1.0 start E:\Dev\web\trtc-electron-simple-demo > concurrently "npm run watch" "npm run electron"	E
[0] [0] > trtc-electron-simple-demo@0.1.0 watch E:\Dev\web\trtc-electron-simple-demo [0] > vue-cli-service buildwatchTRTC_ENV development [0]	
[1] [1] > trtc-electron-simple-demo@0.1.0 electron E:\Dev\web\trtc-electron-simple-demo [1] > electron .	
<pre>[L] [0] process.argv: [[0] 'D:\\nodejs\\node.exe', [0] 'E:\\Dev\\web\\trtc-electron-simple-demo\\node_modules\\@vue\\cli-service\\bin\\vue-cli-service.js [0] 'build', [0] 'watch', [0] '-TRTC_ENV', [0] 'development' [0]] [0] [0]</pre>	5',
[0] [0] getArgvToObject param: { WATCH: 'TRTC_ENV', TRTC_ENV: 'development' } [0] param: { WATCH: 'TRTC_ENV', TRTC_ENV: 'development' } [1] [1] (electron) The default value of app.allowRendererProcessReuse is deprecated, it is currently "false"	'. It
will change to be "true" in Electron 9. For more information please check https://github.com/electron/e n/issues/18397	electro
[0] - Building for development	
	+

プロジェクトのメインコマンド

コマンド	説明
npm run start	開発環境でDemoを実行
npm run pack:mac	Macの.dmgインストールファイルのパッケージング
npm run pack:win64	Windows 64ビットの.exe インストールファイルのパッケージング

よくあるご質問

キーをクエリーするとき、パブリックキーとプライベートキーの情報しか取得できませんが、キーはどうしたら取得できますか。

TRTC SDK 6.6バージョン(2019年08月)では、新しい署名アルゴリズムのHMAC-SHA256の使用を開始しました。それ以前に作成されたアプリケーションの場合、新しい暗号化鍵を取得するために、署名アルゴリズムをアップグレードする必要があります。アップグレードしない場合でも、旧バージョンアルゴリズムECDSA-SHA256を引き続き使用できます。

アップグレード操作:

- 1. TRTCコンソールにログインします。
- 左側ナビゲーションバーでアプリケーション管理を選択し、ターゲットアプリケーションのある行のアプリケーションにある行のアプリケーション情報をクリックします。
 - 3. **クイックマスター**タブを選択して**ステップ2 UserSigを発行するためのキーを取得**エリアの**ここをクリッ クしてアップグレード**をクリックします。

2.2台のデバイスで同時に**Demo**を実行しているのに、お互いの画面が表示されないのはなぜですか。

2台のデバイスでDemoを実行するときは、異なるUserIDを使用していることを確認してください。TRTCでは、 2台のデバイスでの同一UserID(SDKAppIDが異なる場合を除く)の同時使用をサポートしていません。

3. ファイアウォールにはどのような制限がありますか。

SDKがUDPプロトコルを使用してオーディオ・ビデオ伝送を行っているため、UDPをブロックするオフィスネットワークでは使用できません。類似した問題がおありの際は、企業ファイアウォール制限の対応をご参照の上、 問題及び原因解決にお役立てください。

説明:

その他の関連するご質問については、Electronについてのよくあるご質問をご参照ください。

技術的なお問い合わせ

詳細な情報については<mark>お問い合わせ</mark>までご連絡ください。

参考ドキュメント

- SDK APIマニュアル
- SDK更新ログ
- Simple Demoソースコード
- API Exampleソースコード
- Electronについてのよくあるご質問

跑通Demo(Flutter)

最終更新日::2022-04-06 17:09:17

ここでは、主にTRTC Demo(Flutter)を素早く実行する方法について説明します。

注意:

現在、Flutter SDKはAndroidとiOSのみ対応しています。

環境要件

- Flutter 2.0以降のバージョン。
- Android端末向け開発:
 - 。 Android Studio 3.5以降のバージョン。
 - 。 AppにはAndroid 4.1以降のバージョンのデバイスが必要です。
- ・ iOS & macOS端末向け開発:
 - 。 Xcode 11.0以降のバージョン。
 - 。 osxシステムには10.11以降のバージョンが必要です。
 - 。 プロジェクトが有効な開発者による署名を設定済みであることを確認してください。
- Windows 端末向け開発:
 - 。 OS: Windows 7 SP1以降のバージョン(x86-64に基づく64ビットOS)。
 - ディスク容量:IDEと一部のツールのインストールに必要な容量を除く、少なくとも1.64 GB以上の空き容量を確保するようにしてください。
 - 。 Visual Studio 2019をインストールします。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

ステップ1:アプリケーションの新規作成

1. TRTCコンソールにログインし、【開発支援】>【Demoのクイック実行】を選択します。

- 【アプリケーションの作成】をクリックし、TestTRTC などのアプリケーション名を入力します。すでにアプリケーションを作成している場合、【既存のアプリケーションを選択】をクリックします。
- 3. 実際の業務ニーズに応じてタグを追加または編集し、【作成】をクリックします。

1 Create Ap	oplication >	Download Source Code	> (3	Modify Configuration	>	4 Compile and Run
Application Type	O New Cxisting					
Application Name	TestTRTC					
Tag	Tags allow you to manage + Add	resources by category. If existi	ng tags do not m	eet your requirements, you	can manage tag	s here 🛂 .
Create	Reset					

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができます。長さは15文字以内とします。
- タグはTencent Cloudのさまざまなリソースを識別して管理するために使用されます。たとえば、企業に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、企業はTRTCアプリケーションにタグを追加することで部門情報にマークを付けることができます。タグは入力必須ではありません。実際のビジネスニーズに応じてタグを追加または編集できます。

ステップ2:SDKおよびDemoソースコードのダウンロード

1. 実際の業務ニーズに応じて、SDKと関連のDemoソースコードをダウンロードします。



タワンロード完了後	、【タウンロード済み。次へ】をクリックします。
Create Application	Download Source Code Modify Configuration Compile and Run
Download SDK and A	uxiliary Demo Source Code
Platform	Operation
iOS	Download at GitHub Download at Gitee Download Zip
Android	Download at GitHub Download at Gitee Download Zip
Web	Download at GitHub Download at Gitee Download Zip
MacOS	Download at GitHub Download at Gitee Download Zip
Electron	Download at GitHub Download at Gitee Download Zip
Windows	Download at GitHub Download at Gitee Download Zip
Flutter	Download at GitHub
Next Previous	

2.

ステップ3:Demoプロジェクトファイルの設定

- 1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づいて対応する開発環境を選択します。
- 2. TRTC-Simple-Demo/example/lib/debug/GenerateTestUserSig.dart ファイルを見つけて開きます。
- 3. GenerateTestUserSig.dart ファイル内の関連パラメータを設定します:
 - 。 SDKAPPID: デフォルト値はPLACEHOLDERです。実際のSDKAppIDに設定してください。
 - 。 SECRETKEY: デフォルト値はPLACEHOLDERです。実際のキー情報に設定してください。

Create Application > Ownload Source > Code	3 Modify Configuration	>	4 Compile & Ru	n			
Paste SDKAppID and Secret Key to	Decompress the source bug/GenerateTestUser	e package downlo Sig.java File	aded in Step 2, and op	en Android/TRTCSce	nesDemo/d	lebug/src/main/java/o	com/tencent/liteav/c
Specified Location	Android	iOS&macOS	Windows(C++)	Windows(C#)	Web	Mini Program	Electron
-	public class Genera	ateTestUserSig {					
SDKAppID Copy							
	private static	final int SDKAP	PID = 0;				
Secret Key Copy							
b	private static	final int EXPIR	ETIME = 604800;				
* The secret key is sensitive information. Please do not disclose it.							
	private static	final String SE	CRETKEY = "";				
Pasted and Next							

4. 貼り付け完了後、【貼り付け済み。次へ】をクリックすれば、作成が完了します。

5. コンパイル完了後、【コンソールの概要ページに戻る】をクリックすればOKです。

説明:

- ここで言及したUserSigの新規作成ソリューションでは、クライアントコードでSECRETKEYを設定します。この手法のうちSECRETKEYは逆コンパイルによって逆向きにクラッキングされやすく、キーがいったん漏洩すると、攻撃者はTencent Cloudトラフィックを盗用できるようになります。そのためこの方法は、ローカルのDemoクイック実行および機能デバッグにのみ適合します。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appから業務サーバーにリクエストを発出し動的にUserSigを 取得します。詳細はUserSigに関するご質問をご参照ください。

ステップ4:コンパイル実行

- 1. flutter pub get を実行します。
- 2. コンパイルを実行し、デバッグを行います。
 - 。 Android 端末
 - 。 iOS 端末
 - 。 Windows 端末

- 。 macOS 端末
- i. flutter run を実行します。
- ii. Android Studio(3.5およびそれ以降のバージョン)を使用して、ソースプロジェクトを開き、【実行】を クリックすれば完了です。

よくあるご質問

TRTCのログはどのように確認しますか?

TRTCログは、デフォルトで圧縮および暗号化され、接尾辞は .xlog です。アドレスは次のとおりです。

- iOS 端末:sandbox の Documents/log 。
- Android 端末:
 - 。 6.7とそれ以前のバージョン: /sdcard/log/tencent/liteav
 - 。 6.8以降のバージョン: /sdcard/Android/data/パッケージ名/files/log/tencent/liteav/

iOSでビデオが表示できない(Androidは正常)場合はどうすればよいですか?

お客様のプロジェクトの info.plist の io.flutter.embedded_views_preview の値がYESになっていることを確認 してください。

Android Manifest merge failedというコンパイルエラーが発生した場合はどうすればよいで すか??

/example/android/app/src/main/AndroidManifest.xml ファイルを開いてください。

- 1. xmlns:tools="http://schemas.android.com/tools" をmanifestの中に追加します。
- 2. `tools:replace="android:label"をapplicationの中に追加します。

android >	app > src > main > 🔈 AndroidManifest.xml
1 <	manifest xmlns:android=" <u>http://schemas.android.com/apk/res/android</u> "
2	<pre>xmlns:tools="http://schemas.android.com/tools"</pre>
3	<pre>package="com.example.mlp"></pre>
4	<pre><!-- io.flutter.app.FlutterApplication is an android.app.Application that</pre--></pre>
5	calls FlutterMain.startInitialization(this); in its onCreate method.
6	In most cases you can leave this as-is, but you if you want to provide
7	additional functionality it is fine to subclass or reimplement
8	FlutterApplication and put your custom class here>
9	<application< td=""></application<>
10	<pre>tools:replace="android:label"</pre>
11	android:name="io.flutter.app.FlutterApplication"
12	android:label="mlp"
13	android:icon="@mipmap/ic_launcher">



説明:

詳細については、Flutterに関するご質問をご参照ください。

Demoクイックスタート(Unity)

最終更新日::2022-03-30 18:20:36

このサンプルプロジェクトはUnity内でTRTC SDKのクイックインテグレーションを行い、ゲーム中のオーディオ ビデオ通話を実現する方法を示しています。

このサンプルプロジェクトには以下の機能が含まれています。

- 通話の参加および通話からの退出。
- ビデオレンダリングのカスタマイズ。
- デバイス管理、音楽の特殊効果および声の特殊効果。

説明:

- 具体的なAPI機能パラメータの説明については、Unity API概要をご参照ください。
- Unityの推奨バージョン: 2020.2.1f1c1。
- 現在、Android、iOS、Windows、Mac(Macはベータ版テスト中です)プラットフォームをサポート しています。
- Android Build Support 、 iOS Build Support 、 Winodows Build Support および MacOs Build Support モジュールが含まれている必要があります。
- その内、iOS端末の開発には以下が必要です。
 - 。 Xcode 11.0およびそれ以降のバージョン。
 - 。 プロジェクトが有効な開発者による署名を設定済みであることを確認してください。

サンプル実行の順序

手順1:アプリケーションの新規作成

1. TRTCコンソールにログインし、【開発支援】>【Demoクイックスタート】を選択します。

2. 【今すぐ開始】をクリックし、 TestTRTC などのアプリケーション名を入力して、【アプリケーションの作 成】をクリックします。

手順2: SDKとソースコードのダウンロード

1. 自分の実際の業務ニーズに応じて、SDKおよび付属するDemoソースコードをダウンロードします。

2. ダウンロード完了後、【ダウンロードしました。次のステップ】をクリックします。(直接Unityを使用してこのプロジェクトを開くことができます。直接SDKファイルを使用する場合は、SDKパッケージ内

の TRTCUnitySDK/Assets/TRTCSDK/SDK フォルダを自分のプロジェクトAssetsディレクトリ下にコピーすることができます)

- 3. Assets/TRTCSDK/Demo/Tools/GenerateTestUserSig.cs ファイルを見つけて開きます。
- 4. GenerateTestUserSig.cs ファイルの関連パラメータを設定します。
 - 。SDKAPPID:デフォルトは0。実際のSDKAppIDを設定してください。
 - 。 SECRETKEY: デフォルトは空文字列。実際のキー情報を設定してください。

Paste SDKAppID and Secret Key to	Decompress the source package downloaded in Step 2, and open Android/TRTCScenesDemo/debug/src/main/java/com/tencent
Specified Location	bug/GenerateTestUserSiggava File Android iOS&macOS Windows(C++) Windows(C#) Web Mini Program Electron
Specifica Location	public class GenerateTestUserSig {
SDKAppID Copy	
	private static final int EDKAPPID = 0;
Secret Key Copy	
b	private static final int EXPIRETIME = 604800;
* The secret key is sensitive information. Please do not disclose it.	
	private static final String SECRETKEY = "";

手順3:コンパイル実行

- Android プラットフォーム
- iOS プラットフォーム
- Windows プラットフォーム
- macOS プラットフォーム



1. Unity Editorを設定して、【File】>【Build Setting】をクリックし、Androidに切り替えます。

	Build Settings	
Build Settings		:
Scenes In Build ✓ TRTCSDK/Demo/HomeScene ✓ TRTCSDK/Demo/AudioApiTest ✓ TRTCSDK/Demo/RoomSceme		0 1 2
Platform		Add Open Scenes
PC, Mac & Linux Standalone	Android	
ios ios	Texture Compression	Don't override
🚔 Android	ETC2 fallback Export Project	32-bit
tvos tvos	Symlink Sources Build App Bundle (Google Play	/
PJA PS4	Create symbols.zip Run Device	Default device 🔻 Refresh
🐼 Xbox One	Development Build	
WebGL	Autoconnect Profiler Deep Profiling	
	Script Debugging Scripts Only Build	Patch Patch And Run
	Compression Method	LZ4
		Learn about Unity Cloud Build
Player Settings	В	uild Build And Run

- 2. Androidの実機に接続して、【Build And Run】をクリックすると、Demoを実行できます。
- インターフェーステストでは、まずenterRoomの呼び出しをクリックしてから他の関連テストを実行します。 データ表示ウィンドウには呼び出しのクリックが成功したことが表示され、もう1つのウィンドウには呼び出し た情報が表示されます。

Demoサンプル



Demo内にはアップロード済みのAPIの大部分が含まれており、テストおよび参考のために呼び出すことができま す。APIドキュメントについてはSDK API(Unity)をご参照ください。

説明:

UIは部分的に調整され更新される可能性があります。最新バージョンを基準としてください。

ディレクトリ構造

Demoクイックスタート(React Native)

最終更新日::2022-04-02 11:41:25

ここでは、主にTencent CloudのTRTC Demo(React Native)を素早く実行する方法をご紹介します。

環境要件

- ReactNative 0.63およびそれ以降のバージョン
- Node & Watchman。nodeバージョンはv12以上が必要です。
- Android端末向け開発:
 - 。 Android Studio 3.5およびそれ以降のバージョン。
 - 。 AppにはAndroid 4.1およびそれ以降のバージョンのデバイスが必要です。

iOS & macOS端末向け開発:

- Xcode 11.0およびそれ以降のバージョン。
- osxシステムには10.11およびそれ以降のバージョンが必要です。
- プロジェクトが有効な開発者による署名を設定済みであることを確認してください。
- 環境構築については、公式ドキュメントをご参照ください。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

手順1:アプリケーションの新規作成

- 1. TRTCコンソールにログインし、開発支援>Demoクイックスタート**を選択します。
- アプリケーションの作成をクリックし、TestTRTCなどのアプリケーション名を入力します。すでにアプリケーションを作成している場合、既存のアプリケーションを選択**をクリックします。



2. 実際の業務ニーズに応じてタグを追加または編集し、作成をクリックします。

1 Create Ap	plication >	2 Download Source Code	> (3 Modify Configuration	>	4 Compile and Run
Application Type	O New O Exist	ing				
Application Name	TestTRTC					
Tag 🛈	Tags allow you to ma + Add	anage resources by category. If existi	ing tags do not r	neet your requirements, you c	an manage ta	ags here 🖸 .
Create	Reset					

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15文 字以内とします。
- タグはTencent Cloudのさまざまなリソースを識別して管理するために使用されます。たとえば、企業 に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、企業はTRTCアプリケ ーションにタグを追加することで部門情報にマークを付けることができます。タグは入力必須ではあり ません。実際のビジネスニーズに応じてタグを追加または編集できます。

ステップ2:SDKおよびDemoソースコードをダウンロード

1. 実際の業務のニーズに応じて、SDKと付属のDemoソースコードをダウンロードします。

2. ダウンロード完了後、**ダウンロードしました。次のステップ**をクリックします。

注意:

現在、コンソールからはReactNative Demoをダウンロードできません。**直接上記のリンクからDemoソ** ースコードをダウンロードしてください。

ステップ3: Demo プログラムファイルの設定

- 1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づき、対応する開発環境を選択します。
- 2. /debug/config.js ファイルを見つけて開きます。

- 3. SDKAPPID と SECRETKEY パラメータを設定します。
 - 。 SDKAPPID: デフォルトはPLACEHOLDER、実際のSDKAppIDを設定してください。
 - 。 SECRETKEY: デフォルトはPLACEHOLDER、実際のキー情報を設定してください。
- 4. 貼り付け完了後、貼り付けました。次のステップをクリックすれば、作成が完了します。

5. コンパイル完了後、コンソール概要に戻るをクリックすれば終了です。

説明:

- ここで言及したUserSigの新規作成ソリューションでは、クライアントコードでSECRETKEYを設定します。この手法のうちSECRETKEYは逆コンパイルによって逆向きにクラッキングされやすく、キーがいったん漏洩すると、攻撃者はTencent Cloudトラフィックを盗用できるようになります。そのためこの 手法は、ローカルのDemoクイックスタートおよび機能デバッグにのみ適合します。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appから業務サーバーにリクエストを発出し動的にUserSigを 取得します。詳細はUserSigに関するご質問をご参照ください。

ステップ4:権限の設定

実行するにはAppの権限設定が必要です。

- Android端末
- iOS端末
- 1. AndroidManifest.xml の中でAppの権限を設定します。TRTC SDKには以下の権限が必要です。

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature android:name="android.hardware.camera" />
</uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature android:name="android.hardware.camera" />
</uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature android:name="android.hardware.camera.autofocus" />
</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-feature</uses-featu
```

注意

android:hardwareAccelerated="false" は設定しないでください。ハードウェアアクセラレーションを 無効にすると、相手側のビデオストリームがレンダリングできなくなります。

2. Android端末のオーディオ/ビデオの権限は手動でリクエストする必要があります。

```
if (Platform.OS === 'android') {
await PermissionsAndroid.requestMultiple([
PermissionsAndroid.PERMISSIONS.RECORD_AUDIO, //オーディオが必要
PermissionsAndroid.PERMISSIONS.CAMERA, //ビデオが必要
]);
}
```

手順5:コンパイル動作

Metroを起動し、React Nativeプロジェクトディレクトリ下で npx react-native start を実行します。

- Android端末
- iOS端末

新しいウィンドウが開き、開発/デバックが起動します。

npx react-native run-android

Unreal Engine

最終更新日:::2022-03-30 18:10:12

ここでは、主にTencent CloudのTRTC Demo(Unreal Engine)を素早く実行する方法をご紹介します。

注意: 現時点ではWindows、MacOs、ios、Androidをサポートしています。

環境要件

- Unreal Engine 4.27.1およびそれ以降のバージョンを推奨します。
- Android端末向け開発:
 - 。 Android Studio 4.0およびそれ以降のバージョン。
 - 。 Visual Studio 2017 15.6以上のバージョン。
 - 。 実機デバッグのみサポートしています

iOS & macOS端末向け開発:

- Xcode 11.0およびそれ以降のバージョン。
- osxシステムには10.11およびそれ以降のバージョンが必要です。
- プロジェクトが有効な開発者による署名を設定済みであることを確認してください。
- Windows 端末向け開発:
 - 。 OS:Windows 7 SP1およびそれ以降のバージョン(x86-64に基づく64ビットOS)。
 - ディスク容量:IDEと一部のツールのインストールに必要な容量を除く、少なくとも1.64 GB以上の空き容量を確保するようにしてください。
 - 。 Visual Studio 2019をインストールします。

前提条件

Tencent Cloudアカウントの登録を行い、実名認証が完了済みであること。

操作手順

手順1:アプリケーションの新規作成

- TRTCコンソールにログインし、【開発支援】>【快速跑通Demo】を選択します。
 【アプリケーションの作成】をクリックし、TestTRTC などのアプリケーション名を入力します。すでにア
 - プリケーションを作成している場合、【既存のアプリケーションを選択】をクリックします。
- 2. 実際の業務ニーズに応じてタグを追加または編集し、【作成】をクリックします。

1 Create Ap	plication > (2) Download Source Code	> (3) Modify Configuration	> (4) Compile and Run
Application Type	O New Existing		
Application Name	Name your demo		
Tag 🛈	Tags allow you to manage resources by category. If ex + Add	isting tags do not meet your requirements, you ca	an manage tags here .
Create	Reset		

説明:

- アプリケーション名には、数字、中国語と英語の文字、アンダーラインのみを含めることができ、15文 字以内とします。
- タグはTencent Cloudのさまざまなリソースを識別して管理するために使用されます。たとえば、企業 に複数の事業部門があり、各部門に1つ以上のTRTCアプリケーションがある場合、企業はTRTCアプリケ ーションにタグを追加することで部門情報にマークを付けることができます。タグは入力必須ではあり ません。実際のビジネスニーズに応じてタグを追加または編集できます。

ステップ2:SDKおよびDemoソースコードをダウンロード

- 1. 実際の業務ニーズに応じてSDKおよび付属するDemoソースコードをダウンロードします(ご質問があればQQ グループ番号:764231117に参加してお問い合わせください)。
- 2. ダウンロード完了後、【ダウンロード済み。次へ】をクリックします。

ステップ3: Demo プログラムファイルの設定

- 1. 設定変更画面に進み、ダウンロードしたソースコードパッケージに基づき、対応する開発環境を選択します。
- 2. /TRTC_Demo/Source/debug/include/DebugDefs.h のファイルを見つけて開きます。
- 3. DebugDefs.h のファイルの関連パラメータを設定します。

- 。 SDKAPPID: デフォルトは0。実際のSDKAppIDを設定してください。
- 。 SECRETKEY: デフォルトは""。実際のキー情報を設定してください。

Create Application > Create Application > Code	3 Modify > (4) Compile & Run
Paste SDKAppID and Secret Key to Specified Location	Decompress the source package downloaded in Step 2, and open Android/TRTCScenesDemo/debug/src/main/java/com/tencent/liteav/de bug/GenerateTestUserSig.java File Android iOS&macOS Windows(C++) Windows(C#) Web Mini Program Electron White place GenerateTestUserSig. (
SDKAppID Copy	private static final int SDRAPPID = 0;
Secret Key Copy b	private static final int EXPIRETIME = 604800;
* The secret key is sensitive information. Please do not disclose it.	
Pasted and Next	private static final String SECRETKEY = "";

4. 貼り付け完了後、【貼り付けました。次へ】をクリックすれば、作成が完了します。

5. コンパイル完了後、【コンソールの概要ページに戻る】をクリックすればOKです。

説明:

- ここで言及したUserSigの新規作成ソリューションでは、クライアントコードでSECRETKEYを設定します。この手法のうちSECRETKEYは逆コンパイルによって逆向きにクラッキングされやすく、キーがいったん漏洩すると、攻撃者はTencent Cloudトラフィックを盗用できるようになります。そのためこの 手法は、ローカルのDemoクイックスタートおよび機能デバッグにのみ適合します。
- UserSigの正しい発行方法は、UserSigの計算コードをサーバーに統合し、Appのインターフェース向け に提供します。UserSigが必要なときは、Appから業務サーバーにリクエストを送信し、動的に UserSigを取得します。詳細はサーバーでのUserSig新規作成をご参照ください。

手順4:コンパイルとパッケージ化の実行

- 1. /TRTC_Demo/TRTC_Demo.uproject をダブルクリックして開きます。
- 2. コンパイルを実行し、デバッグを行います。
 - 。 macOS 端末

- 。 Windows 端末
- 。 iOS 端末
- 。 Android 端末
- i. File -> Package Project -> Mac
- ii. 権限を設定します。前の手順でコンパイルしたxxx.appファイルを右クリックし、「パッケージの内容を表示」を選択します



- iii. 「Contents->Info.plist」に進みます
- iv. 「Information Property List」を選択し、次の2つの権限を追加します。

<key>NSCameraUsageDescription</key> <string>通常のビデオ通話が行えるようにカメラを許可します</string> <key>NSMicrophoneUsageDescription</key> <string>通常の音声通話が行えるようにマイクの権限を承認します</string>

v. UE4のeditorで実行中の場合は、UE4Editor.appファイルを見つけて、上記の手順に従って権限を追加す る必要があります。

Demoの実行

Demoでは1対1のビデオ通話が実現でき、テストおよび参考のために呼び出すことができます。APIドキュメント についてはC++ 全プラットフォームAPIをご参照ください。

説明:


UIは部分的に調整され更新される可能性があります。最新バージョンを基準としてください。



TRTC全プラットフォーム C++ APIドキュメント

中国語ドキュメント

英語ドキュメント

よくあるご質問

TRTCのログはどうやって確認しますか。

TRTCログは、デフォルトで圧縮および暗号化され、接尾辞は .xlog です。アドレスは次のとおりです。

- iOS 端末: sandbox の Documents/log 。
- Android 端末:

- 。 6.7とそれ以前のバージョン: /sdcard/log/tencent/liteav
- 。 6.8以後のバージョン: /sdcard/Android/data/パッケージ名/files/log/tencent/liteav/

macos UE4 editorのクラッシュ

UE4Editor.app info.plistでオーディオとビデオの権限を設定しているかを確認してください

<key>NSCameraUsageDescription</key> <string>通常のビデオ通話が行えるようにカメラを許可します</string> <key>NSMicrophoneUsageDescription</key> <string>通常の音声通話が行えるようにマイクの権限を承認します</string>

Androidでの「Attempt to construct staged filesystem reference from absolute path"」エラー

UE4プロジェクトを閉じ、cmdを開きます

adb shell cd sdcard ls (you should see the UE4Game directory listed) rm -r UE4Game プロジェクトを再コンパイルします

入門者のよくあるご質問

最終更新日::2022-02-18 15:16:10

UserSigとは。

UserSigとは、悪意ある攻撃者によるクラウドサービスの使用権の盗用を防ぐために、Tencent Cloudによって設計されたセキュリティ保護された署名です。

現在、TRTC、IMおよびMLVBなどのサービスには、すべてこの一連のセキュリティ保護メカニズムが採用されて います。これらのサービスを使用する場合、対応するSDKの初期化またはログイン関数において、SDKAppID、 UserIDおよびUserSigという3つの重要情報を提供する必要があります。

このうちSDKAppIDはお客様のアプリケーションの識別に、UserIDはユーザーの識別に使用されます。UserSig は、** HMAC SHA256 **暗号化アルゴリズムによって算出される最初の2つをもとに計算されたセキュリティ署 名です。攻撃者がUserSigを偽造できない限り、クラウドサービスのトラフィックを盗用することはできません。 UserSigの計算原理を次の図に示します。SDKAppID、UserID、ExpireTimeなどの重要情報をハッシュ化・暗号 化することがUserSigの本質です。

//UserSigの計算式。このうちsecretkeyは、usersigを算出するための暗号化鍵です。 usersig = hmacsha256(secretkey, (userid + sdkappid + currtime + expire + base64(userid + sdkappid + currtime + expire)))

説明:

- currtime は現在のシステムの時間で、 expire は署名の期限切れの時間です。
- UserSigの具体的な計算取得方法を知りたい場合は、 UserSigの詳細な説明をご参照ください。

TRTCではルームを同時にいくつ作成できますか。

4,294,967,294室のルームの同時併存をサポートします。ルームの累計数は無制限です。

TRTCの遅延はどのくらいですか。

グローバルなエンドツーエンドの平均遅延は300ms未満です。

TRTCはPCでの画面共有をサポートしていますか。

サポートしています。次のドキュメントをご参照ください。

- 画面共有(Windows)
- 画面共有(Mac)
- 画面共有(Web)

画面共有インターフェースの詳細についてはWindows(C++)APIをご参照ください。また、Electronインター フェースもご利用いただけます。

TRTCはどのプラットフォームをサポートしますか。

サポートしているプラットフォームは、iOS、Android、Windows(C++)、Windows(C#)、Mac、Web、 Electronです。詳細については、プラットフォームのサポートをご参照ください。

TRTCは最大何人の同時通話をサポートできますか。

- 通話モードでは、1ルームあたり最大300人の同時接続、最大50人のカメラまたはマイクの同時使用をサポートしています。
- ライブストリーミングモードでは、1ルームあたり視聴者10万人のオンライン視聴と、キャスター50人のカメ ラまたはマイクの使用をサポートしています。

TRTCはライブストリーミングのシーン類のアプリケーションをどのように実現しますか。

TRTCは、特にオンラインライブストリーミングのシーン向けに、10万人向けの低遅延ライブストリーミングソリ ューションをリリースしました。これにより、キャスターとマイク接続したキャスター間の最小遅延が200ms、 通常の視聴者の遅延が1秒以内になるだけでなく、脆弱なネットワークに極めて高い耐性を持つこととなり、モバ イル端末の複雑なネットワーク環境に対応します。

操作ガイドの詳細はライブストリーミングクイックスタートモードをご参照ください。

TRTCライブストリーミングはどのようなロールをサポートしていますか。ロールの違いは何ですか。

ライブストリーミングのシーン(TRTCAppSceneLIVEとTRTCAppSceneVoiceChatRoom)は、 TRTCRoleAnchor(キャスター)とTRTCRoleAudience(視聴者)という2つのロールをサポートしています。 違いとしては、キャスターのロールはオーディオ・ビデオデータのアップロードとダウンロードを同時に行うこ とができますが、視聴者のロールは他人のデータのダウンロードと再生のみをサポートしていることです。 switchRole()を呼び出すことにより、ロールを切り替えることができます。

TRTCルームは、キックアウト、発言の禁止、ミュートをサポートしていますか。

サポートしています。

- 簡単なシグナリング操作の場合、TRTCのカスタムシグナルインターフェースsendCustomCmdMsgを使用できます。開発者が対応する制御シグナリングをカスタマイズして、制御シグナリングを受信した通話者が対応する操作を実行すればOKです。例えば、キックアウトとは、追い出しシグナリングを定義することであり、この信号を受信したユーザーは自動的に退室します。
- より完全な操作ロジックを実行する必要がある場合は、開発者がIMを使用して関連のロジックを実行し、TRTC ルームとIMグループとのマッピングを行い、IMグループでカスタムメッセージを送受信して、対応する操作を 実行することをお勧めします。

TRTCオーディオ・ビデオストリームは、CDNを介したプルストリームによる視聴をサポートして いますか。

サポートしています。詳細については、CDN relayed live streamingの実装をご参照ください。

iOS端末はSwiftの統合をサポートしていますか?

サポートしています。サードパーティライブラリのフローにそのまま従ってSDKを統合すればOKです。また、 Demoクイックスタート(iOS&Mac)も参照することができます。

デスクトップブラウザSDKはどのブラウザをサポートしていますか。

現在、デスクトップ版Chromeブラウザ、デスクトップ版Safariブラウザおよびモバイル版Safariブラウザのサポ ート状態は比較的万全です。その他のプラットフォーム(Androidプラットフォームのブラウザなど)のサポート 状態はまだ不十分です。詳細については、サポートするプラットフォームをご参照ください。

ブラウザでWebRTC能力テストを開き、WebRTC機能を完全にサポートしているかテストすることができます。

Web端末SDKログのエラーメッセージのうち、NotFoundError、NotAllowedError、 NotReadableError、OverConstrainedErrorおよびAbortErrorは、それぞれどういう意味 ですか。

エラー名	説明	推奨する対処方法
NotFoundError	リクエストを満たすパラメー タのメディアタイプ(オーデ ィオ、ビデオ、画面共有を含 む)が見つかりません。 例えば、PCにカメラがないの に、ブラウザにビデオストリ ームを取得するようリクエス トがあった場合、このエラー が発生します。	ユーザーが通話を開始する前に、通話に必要 なカメラやマイクなどのデバイスを確認する ことをお勧めします。カメラがなく、音声通 話を行う必要がある場合は、 TRTC.createStream({ audio: true, video: false })で、マイクのみをキャプチャするよ うに指定できます。
NotAllowedError	ユーザーが、現在のブラウ ザ・インスタンスのオーディ オ、ビデオおよび画面共有へ のアクセスのリクエストを拒 否しました。	ユーザーに対し、カメラ/マイクへのアクセ ス権限を承認しないと、オーディオビデオ通 話を行うことができません、というプロンプ トが表示されます。
NotReadableError	権限が付与されたユーザーが 対応するデバイスを使用して いますが、OS上のいずれかの ハードウェア、ブラウザまた はWebページの階層に発生し たエラーのため、デバイスに アクセスできません。	ブラウザのエラーメッセージに従って処理す ると、ユーザーに対し、「現在カメラ/マイ クにアクセスできません。他のアプリケーシ ョンがカメラ/マイクへのアクセスをリクエ ストしていないことを確認してから、もうー 度お試しください」というプロンプトが表示 されます。

エラー名	説明	推奨する対処方法
OverConstrainedError	camerald/microphoneldのパ ラメータの値が無効です。	camerald/microphoneldの渡された値が正 しく有効であることを確認してください。
AbortError	何らかの理由により、デバイ スを使用できません。	-

詳細については、initializeをご参照ください。

Web端末SDKがデバイス(カメラ/マイク)リストを正常に取得できるかどうか確認するにはどう すればいいですか。

1. ブラウザがデバイスを正常に使用できるかチェックします。

ページ上でコンソールを直接開き、 navigator.mediaDevices.enumerateDevices() と入力して、デバイスリス トを取得できるかどうか確認します。

- デバイスを正常に取得できるとPromiseが返ってきて、その中に MediaDeviceInfoのオブジェクトの配列があります。配列の中の各オブジェクトが1つの使用可能なメディアデバイスに対応します。
- 列挙に失敗すると、Promiseは rejectedを戻し、これは、ブラウザがデバイスを識別できなかったことを表し ます。ブラウザまたはデバイスをチェックする必要があります。
- デバイスリストを取得できる場合は、 navigator.mediaDevices.getUserMedia({ audio: true, video: true
 と入力して、MediaStreamオブジェクトが正常に返されるかどうか確認します。正常に返されない場合、 ブラウザがデータを取得していないことを示しているので、ブラウザの設定をチェックする必要があります。

ライブストリーミング、インタラクティブライブストリーミング、TRTCおよびRelayed live streamingの違いと関係性は何ですか。

- LVB(キーワード:一対多、RTMP/HLS/HTTP-FLV、CDN)
 LVBは、プッシュ端末、再生端末およびライブストリーミングクラウドサービスに分かれます。クラウドサービスはCDNを使用してライブストリームを配信します。プッシュには一般的な標準プロトコルRTMPが使用されます。CDNによって配信された場合、再生するときには通常、RTMP、HTTP-FLVまたはHLS(H5サポート)を選択して視聴することができます。
- ILV(キーワード:マイク接続、PK)
 ILVBは、業務形式の1種で、キャスターと視聴者の間のインタラクティブなマイク接続や、キャスターとキャスターの間のインタラクティブなPKを行うライブストリーミングのタイプの1つです。
- TRTC(キーワード:マルチプレイヤーインタラクション、UDPプライベートプロトコル、低遅延)
 TRTCの主なユースケースは、オーディオとビデオのインタラクションと低遅延のライブストリーミングです。
 UDPベースのプライベートプロトコルを使用し、ディレイは100ミリ秒まで引き下げることができます。典型
 的なシーンは、QQ電話、Tencent Meeting、大規模セミナーなどです。TRTCはプラットフォーム全体をカバ

- ーし、iOS/Android/Windowsに加えて、WebRTCの相互運用性、クラウドミクスストリーミングという方法 で、CDNへの画面のRelayed live streamingもサポートします。
- Relayed live streaming(キーワード:クラウドミクスストリーミング、RTCバイパス・プッシュ転送、 CDN)

Relayed live streamingとは、低遅延のマイク接続ルームにおけるマルチチャンネルのプッシュ画面をコピー して、クラウド内で画面を混合して一つのチャネルにし、ミクスストリーミング後の画面をライブCDNにプッ シュして配信、再生する技術のことです。

TRTCは、通話時間と使用量をどのように確認すればいいですか。

TRTCコンソールの【使用量の統計】ページで確認することができます。

TRTCにラグが発生した場合はどのように調査すればいいですか。

通話品質は、対応するRoomIDとUserIDを用いて、TRTCコンソールの【監視ダッシュボード】ページで確認する ことができます。

- 受信端末の視点から送信端末と受信端末ユーザーの状況を確認します。
- ・送信端末と受信端末のパケット損失率が高いかどうか確認します。パケット損失率が通常より高い場合、ネットワーク状態が不安定なためにラグが発生しています。
- フレームレートとCPU使用率を確認します。フレームレートが低くCPU使用率が高いと、ラグが発生します。

TRTCに画質の不良、ぼやけ、モザイクなどが発生する場合はどのように調査すればいいですか。

- 解像度は主にビットレートに関係しています。SDKのビットレートが低く設定されているか確認してください。解像度が高く、ビットレートが低いとモザイク現象が起こりやすくなります。
- TRTCは、クラウドQOSトラフィックコントロールポリシーを通じて、ネットワークの状態に応じ、ビットレートと解像度を動的に調整します。ネットワークが貧弱な場合、ビットレートが下がりやすくなり、解像度が低下します。
- 入室時にVideoCallモードとLiveモードのどちらを使用しているかチェックします。通話シーン向けの VideoCallモードは低遅延とスムーズさの維持に重点を置いています。したがって脆弱なネットワークの場合、 スムーズさを確保するために画質が犠牲になりやすくなります。画質の方が大事なシーンには、Liveモードの 使用をお勧めします。

SDKの最新のバージョン番号はどのように確認しますか。

- 自動ロードを使用する場合は、 latest.release により最新バージョンとマッチングされて自動ロードが実行 されるため、バージョン番号を変更する必要はありません。具体的な統合方法についてはSDKクイックインテ グレーションをご参照ください。
- 現在のSDKの最新バージョン番号はリリースノートから確認することができます。以下をご参照ください。
 - 。 iOS & Androidでは、リリースノート(App)をご参照ください。
 - 。 デスクトップブラウザでは、リリースノート(Web)をご参照ください。

。 Electronでは、リリースノート(Electron)をご参照ください。