Mobile Live Video Broadcasting SDK Download Product Documentation





Copyright Notice

©2013-2022 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.



Contents

SDK Download Download SDK Integration iOS Android Features

Support for RTC Publishing

SDK Download Download

Last updated : 2022-09-15 10:23:19

We offer the MLVB Professional SDK for users outside the Chinese mainland. To learn about its features and the license required to use it, see Features.

MLVB Professional

MLVB Professional integrates multiple core audio/video features of Tencent Cloud, including MLVB, TRTC, live publishing/playback, and basic filters.

Platform	ZIP File	64-bit Support	Increase Installation Package By	Downsizing Installation Package
iOS	Download	Yes	5.69MB (arm64)	Document
Android	Download	Yes	9.15MB (armv7) 10.4MB (arm64)	Document

Note:

Different authorization is required to use different features of MLVB Professional:

- To use MLVB features, you must purchase an MLVB Professional license.
- To use TRTC features, you must purchase a TRTC package.
- Before you use the MLVB SDK, you need to set up the GDPR environment:

Android :

V2TXLivePremier.setEnvironment("GDPR");

iOS :

[V2TXLivePremier setEnvironment:@"GDPR"];

SDK Integration

Last updated : 2022-09-14 17:14:53

This document describes how to quickly integrate RT-Cube's MLVB LiteAVSDK for iOS into your project.

Environment Requirements

- Xcode 9.0 or above
- iPhone or iPad with iOS 9.0 or above
- A valid developer signature for your project

Integrating the SDK

You can use CocoaPods to automatically load the SDK or manually download the SDK and import it into your project.

CocoaPods

1. Install CocoaPods

Enter the following command in a terminal window (you need to install Ruby on your macOS first):

sudo gem install cocoapods

2. Create a Podfile

Go to the directory of your project and enter the following command to create a Podfile in the directory.

pod init

3. Edit the Podfile

There are two ways to edit the Podfile:

• Method 1: use the path of the PODSPEC file of LiteAVSDK

```
platform :ios, '9.0'
target 'App' do
```

```
pod 'TXLiteAVSDK_Professional', :podspec => 'https://liteav.sdk.qcloud.com/po
d/liteavsdkspec/TXLiteAVSDK_Professional.podspec'
end
```

• Method 2: use CocoaPod's official source, which allows version selection

4. Update the local repository and install the SDK

Enter the following command in a terminal window to update the local repository file and install LiteAVSDK:

```
pod install
```

Or, run the following command to update the local repository:

pod update

An XCWORKSPACE project file integrated with LiteAVSDK will be generated. Double-click to open the file.

Manual integration

- 1. Download LiteAVSDK and decompress the file.
- 2. Open your Xcode project, select the target you want to run, and select **Build Phases**.

🔣 TXLiteAVDemo		< 🔺 >
	General Signing & Capabilities Resource Tags Info Build Settings Build Phases Build Rules	
PROJECT	+ Bilter	
🖾 TXLiteAVDemo	> Dependencies (0 items)	
TARGETS	> [CP] Check Pods Manifest.lock	
TXLiteAVDemo_International	> Compile Sources (117 items)	
	> Link Binary With Libraries (0 items)	
	> Copy Bundle Resources (29 items)	
	> Embed App Extensions (0 items)	
	> [CP] Embed Pods Frameworks	
	> [CP] Copy Pods Resources	
	> Embed Frameworks (3 items)	



3. Expand Link Binary with Libraries and click + at the bottom to add the libraries to depend on.

□ IXLiteAVDemo.xcodepr	oj									₹∣₽
🖾 TXLiteAVDemo										< 🔺 >
	General	Signing & Capabilities	Resource Tags	Info	Build Settings	Build Phases	Build Rules			
PROJECT								🕞 Filter		
🛃 TXLiteAVDemo	> Deper	ndencies (0 items)								
TARGETS	> [CP] C	heck Pods Manifest.loci	k							
TXLiteAVDemo_International	> Comp	ile Sources (117 items)								
	∽ Link B	inary With Libraries (0 ii	tems)							
		Name							Status	
					Add fran	neworks & librarie	s here			
		+ -								
	> Copy	Bundle Resources (29 ite	ems)							
	> Embe	d App Extensions (0 item	ns)							
	> [CP] E	mbed Pods Frameworks								
	> [CP] C	opy Pods Resources								
	> Embe	d Frameworks (3 items)								

4. Add the downloaded TXLiteAVSDK_Professional.framework and the libraries it depends on:

libz.tbd
libc++.tbd
libresolv.tbd
libsqlite3.tbd
Accelerate.framework
OpenAL.framework



器 I く > 区 TXLiteAVDemo.xcodepro	oj									₹ 🕀
K TXLiteAVDemo										< 🔺 >
	General	Signing & Capabilities	Resource Tags	Info	Build Settings	Build Phases	Build Rules			
PROJECT -	+							🕞 Filter		
🖾 TXL iteAVDemo										
	> Depe	naencies (U items)								
TARGETS	> [CP] (Check Pods Manifest.loci	ĸ							×
🔁 TXLiteAVDemo_International	> Comr	pile Sources (117 items)								×
	🗸 Link I	Binary With Libraries (7 it	tems)							×
		Name							Status	
		🚔 TXLiteAVS	DK_International.fra	mework					Required \$	
		🖹 libz.tbd							Required 🗘	
		🗐 libc++.tbd							Required 🗘	
		🗐 libresolv.tb	d						Required 🗘	
		🗐 libsqlite3.t	bd						Required 🗘	
		🚔 Accelerate	.framework						Required 🗘	
		👄 OpenAL.fra	amework						Required 🗘	
		+ -			Drag to	reorder linked bina	aries			
	> Copy	Bundle Resources (29 ite	ems)							×
	> Embe	ed App Extensions (0 item	ns)							×
	> [CP] I	Embed Pods Frameworks								×
		Conv Pods Resources								×

5. Click Build Settings, search for Other Linker Flags , and add -ObjC .

∎ Ge	eneral S	igning & Cap	abilities	Resource	e Tags	Info	Build Settin	ngs E	Build Phase	es	Build Rules	
PROJECT		Basic	Customiz	ed All	Co	ombined	Levels	+		(Q~	Other	8
🛃 MLVB-API-Example	e-OC											
TARGETS		✓ Build Op	tions Setting						🔗 MI VB-	-API-F	xample-OC	
🔗 MLVB-API-Example	e-OC		Enable	Bitcode					No \$			
TXReplayKit_Screet	n											
		\sim Linking										
		_	Setting						🔗 MLVB-	-API-E	xample-OC	
			Other L	inker Flags	s				-ObjC			

Granting Camera and Mic Permissions

To use the audio/video features of the SDK, you need to grant it mic and camera permissions. Add the two items below to Info.plist of your application to display pop-up messages asking for mic and camera permissions.

• Privacy - Microphone Usage Description, plus a statement specifying why mic access is needed



· Privacy - Camera Usage Description, plus a statement specifying why camera access is needed

器 < >		₹	
🖾 TXLiteAVDemo $ angle \equiv$ TXLiteAVDemo $ angle \equiv$ Supporting Files $ angle$	\blacksquare Info.plist \rangle	No Selection	< 🔺
Кеу	Туре	Value	
 Information Property List 		(21 items)	
Localization native development region		China	0
Bundle display name		腾讯云视立方	
Executable file		\$(EXECUTABLE_NAME)	
Bundle identifier		\$(PRODUCT_BUNDLE_IDENTIFIER)	
InfoDictionary version		6.0	
Bundle name 🗘		\$(PRODUCT_NAME)	
Bundle creator OS Type code		????	
> URL types		(1 item)	
Bundle version		8167	
IMSDKCrashReporterEnable	Boolean	0	0
> LSApplicationQueriesSchemes	Array	(1 item)	
Application requires iPhone environment		YES	0
> App Transport Security Settings		(1 item)	
Privacy - Camera Usage Description			
Privacy - Microphone Usage Description			
> Required background modes		[1 nemy	
Application supports iTunes file sharing		YES	0
Launch screen interface file base name		LaunchScreen	
> Required device capabilities		(1 item)	
> Supported interface orientations		(1 item)	
View controller-based status bar appearance		NO	0

Importing the SDK

There are two ways to import the SDK in your project code.

• Method 1: import the SDK module in the files that need to use the SDK's APIs in your project

```
@import TXLiteAVSDK_Professional;
```

• Method 2: import a specific header file in the files that need to use the SDK's APIs in your project

#import "TXLiteAVSDK_Professional/TXLiteAVSDK.h"

Configuring License

Log in to the CSS console, go to **MLVB SDK** > License Management, and click **Get License** to obtain a trial license. For detailed directions, see Applying for trial license. You will get two strings: a license URL and a decryption



key.

Before you use LiteAVSDK features in your application, complete the following configurations (preferably in -

```
[AppDelegate application:didFinishLaunchingWithOptions:] ):
```

```
@import TXLiteAVSDK_Professional;
@implementation AppDelegate
- (BOOL)application: (UIApplication *)application didFinishLaunchingWithOptions: (N
SDictionary *)launchOptions {
NSString * const licenceURL = @"<The license URL obtained>";
NSString * const licenceKey = @"<The key obtained>";
//TXLiveBase can be found in the "TXLiveBase.h" header file
[TXLiveBase setLicenceURL:licenceURL key:licenceKey];
NSLog(@"SDK Version = %@", [TXLiveBase getSDKVersionStr]);
}
@end
```

FAQs

Can I run LiteAVSDK in the background?

Yes, you can. If you want the SDK to run in the background, select your project, under the **Capabilities** tab, set **Background Modes** to **ON**, and check **Audio**, **AirPlay and Picture in Picture**, as shown below:

ය 🕹 🔁 TXLiteAVDem	0							< 🔺 >
	General		Resource Tags	Info	Build Settings	Build Phases	Build Rules	
PROJECT								
🛃 TXLiteAVDemo	► < <p>Access WiFi Information</p>							OFF
TARGETS								
🔗 TXLiteAVDemo_Pr	▶ 🕀 App Groups							OFF
	▶ 💼 Apple Pay							OFF
	▶							OFF
	▶ 🔍 AutoFill Credential Provi	der						OFF
	▼ 🕑 Background Modes							ON
		Modes: 🔽 Au	udio, AirPlay, and Pictu	re in Picture	9			
		Lo	cation updates					
		Vc	oice over IP					
		Ne Ne	ewsstand downloads					
		Ex	ternal accessory com	munication				
		U	ses Bluetooth LE acces	sories				
		Ac	ts as a Bluetooth LE a	ccessory				
			ackground tetch					
			smole notifications					

Android

Last updated : 2022-09-14 17:17:02

This document describes how to quickly integrate Tencent Cloud LiteAVSDK for Android into your project.

Environment Requirements

- Android Studio 2.0 or above
- Android 4.1 (SDK API level 16) or above

Integrating the SDK (AAR)

You can use Gradle to automatically load the AAR file or manually download the AAR file and import it into your project.

Method 1: automatic loading (AAR)

Since JCenter has been deprecated, you can configure a Maven Central repository in Gradle to automatically download and update LiteAVSDK.

Open your project with Android Studio and modify the build.gradle file as described below to complete the integration.

LiteAVSDKDemo > app > # build.gradle >		
ष्टु 🔲 Project 👻	🕄 😤 🗢 🚍 🚔 setivity, main.xml × 💿 MainActivity java × 🙀 build gandle (app) ×	
🖉 🗸 📭 LiteAVSDKDemo ~/LiteAVSDKDemo	Could Block have depended since but explore your. A seriest way we assesses for the IPS to much expension	Sure New Jacob Bar
🖬 🔉 🖿 .gradle	visure ines have oranged since last project spire. A project spire may be necessary for one for to work property.	Sync Now Is note on
a > 🖿 .idea	2 id 'com.android.annlication'	
app v 📷 app		
ar S Ban libs		
S > E src		
	6 compileSdkVersion 30	
w build.gradie	7 buildToolsVersion "30.0.3"	
aroguard-rules.pro		
> Im gradie	annicationfd "com tencent liteav liteavsdkdemo"	
A altionare	appleduction of the state of th	
ng igregroot	12 targetSdkVersion 30	
All produce promotion	13 versionCode 1	
gradie, properties	14 versionName "1.0"	
Ball gradiew		
gradiew.bat	16 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"	
local properties	17 th ndk {	
And settings.gradie	appletiters "armeaper", "armeaper-via", "armeaper-via	
Illi External Libraries		
To Scratches and Consoles		
	22 buildTypes {	
	23 deste t	
	24 minifyEnabled false	
	25 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'	
	2/ J F	
	20 sourceConactibility JavaVersion.VERSION 1 8	
	30 targetCompatibility JavaVersion.VERSION 1 8	
	34) dependencies /	
	35 inplementation 'com.tencent.liteav:LiteAVSDK_International:Latest.release' 2	
	in a manufaction i com council a addred in staticial instancial: 4.81	
	indementation 'androids.constraint layout:constraint layout:2.1.2'	
	39 testImplementation 'junit:junit:4.+'	
	40 androidTestImplementation 'androidx.test.ext:junit:1.1.3'	
	androidTestImplementation 'androidx.test.espresso-core:3.4.0'	

- 1. Open build.gradle under your application.
- 2. Add the LiteAVSDK dependency to dependencies .

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

Or

```
dependencies {
implementation 'com.tencent.liteav:LiteAVSDK_Professional:latest.release@aar'
}
```

3. In defaultConfig , specify the CPU architecture to be used by the application. Currently, LiteAVSDK supports armeabi, armeabi-v7a, and arm64-v8a.

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

1

4. Click the **Sync Now** button to sync the SDK. If you have no problem accessing Maven Central, the SDK will be downloaded and integrated into your project automatically.

Method 2: manual download (AAR)

If you have problem accessing Maven Central, you can manually download the SDK and integrate it into your project.

1. Download LiteAVSDK and decompress the file.

- Stencent Cloud
- 2. Copy the AAR file in the SDK directory to the app/libs directory of your project.



3. Add flatDir to build.gradle under your project's root directory to specify the local path for the repository.





4. Add the LiteAVSDK dependency and, in app/build.gradle , add code that references the AAR file.



implementation(name:'LiteAVSDK_Professional_8.7.10102', ext:'aar')

5. In defaultConfig of app/build.gradle, specify the CPU architecture to be used by the application. Currently, LiteAVSDK supports armeabi, armeabi-v7a, and arm64-v8a.

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

6. Click **Sync Now** to complete the integration of LiteAVSDK.

Integrating the SDK (JAR)

If you do not want to import the AAR library, you can also integrate LiteAVSDK by importing JAR and SO libraries.

1. Download LiteAVSDK and decompress the file. In the SDK directory, find

```
LiteAVSDK_Professional_xxx.zip ( xxx indicates the version number of LiteAVSDK).
```

LiteAVSDK_Soid_6.8.7969	SDK	
TXLiteAVSDroid_latest.zip	🖿 API文档	LiteAVSDK_S6.8.7969.zip
🖿 perf	Demo	LiteAVSDK_S6.8.7969.aar

Decompress the file, and you will find a libs directory that contains a JAR file and several SO folders, as shown below:

💼 libs 🔹 🕨	🖿 arm64-v8a	
	🚞 armeabi	
	armeabi-v7a	
	🚆 liteavsdk.jar	

2. Copy the JAR file and armeabi, armeabi-v7a, and arm64-v8a folders to the app/libs directory.



3. Add code that references the JAR library in app/build.gradle .



```
dependencies {
implementation fileTree(dir:'libs',include:['*.jar'])
}
```



4. Add **flatDir** to build.gradle under the project's root directory to specify the local path for the repository.

	• •)													Lite	eAVSD⊧	KDemo – build.gra	adle (LiteAVS	DKDemo)	
1		I G	← →	→ <	🔺 app	▼ No	Devices				<i>0</i> , ₹	x	R [<u>,</u>		▶	۹			
L	iteA	VSDK	Demo $ angle$	🗬 build.	gradle															
ect		Proje	ct 🔻		⊕ ÷	\$ -	– 🔗 t	ouild.gra	dle (:app)	🛛 🔗 bı	ıild.gradl	e (LiteA	VSDKDei	mo) $ imes$						
📕 <u>1</u> : Proj	▼ 	Lite	eAVSDK .gradle .idea	Demo ~				/ / bu	Top-le ildscri repos	evel bui ipt { sitories	ld fil.	e whe.	re you	u can	add c	onfi	.guration o	ptions c	ommon to a	
Resource Manager	•	′ III ► ► 10 ∞	app libs src d .gitig d build gradle .gitigno build.gr	nore .gradle uard-rul re adle	es.pro				} deper	ndencies dencies lasspat // NOTE: // in th	tral() { h "com Do no e indi	ı.andr t pla vidua	oid.ta ce you l modu	ools.b ur app ule bu	uild: licat ild.g	grad ion gradl		es here;	they belo	
	► 2	ili Ext	gradle.p gradlew gradlew local.pro settings ernal Li atches a	ropertie .bat operties .gradle braries .nd Cons	s		14 15 16 17 18 20 21 22 23 24 25 26 26	- - al 	lprojec repos (1) } sk clea delet	sitories poogle() navenCen latDir dirs an(type: se rootP	<pre>{ tral() { 'libs Delet roject</pre>	e) {	dDir							
							27													



5. In app/build.gradle , add code that references the SO libraries.



6. In defaultConfig of app/build.gradle , specify the CPU architecture to be used by the application.

Currently, LiteAVSDK supports armeabi, armeabi-v7a, and arm64-v8a.

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

7. Click **Sync Now** to complete the integration.

Setting Packaging Parameters



😨 PituDemo [E:\MyStudioWorkspace2\PituDe	mo] - app - Android Studio	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e	<u>R</u> efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S W</u> indow <u>H</u> elp	
属 PituDemo 👌 📷 app 👌 📀 build.gradle 👌		
법 Project 🗸 😳 崇	🏶 🗜 🧿 CameraPusherActivity.java 🛛 🚺 app 👋 💿 PusherSettingFragment.java 👋 💿 MainActivity.java 👋 🛔	🛔 activity_main.xml 🗵 🧿 IntentBean.java 🔅
Image: Specific State Image: Specific State Image: Specific State Image: Specific State <	PituDemc 10 Version water 1.0 11 testInstrumentationRunner "android. support. test. runner. AndroidJU 12 multiDexEnabled true	
en ▼ R app to ► I build	13 14 15 D mdk (
iii ► ■ iius iii ► ■ src iii _ gitignore	16 abiFilters "armeabi-v7a" 17 \ominus }	
e app.iml c build.gradle c de	18 } 19 buildTypes { 20 prelease {	
 ► ■ gradle ► ➡ libsuperplayer ► ➡ lib libsuperplayer 	21 minifyEnabled false 22 proguardFiles getDefaultProguardFile('proguard-android.txt')	
 Image player Image player Image player 		
Ini videoediter Ini videojoiner Ini videojoiner Ini videorecorder	26 pickFirst '**/libc++_shared.so' 27 doNotStrip "*/armeabi/libYTCommon.so"	
▶ ∎îîivideouploader ≝.gitignore ⓒ build.gradle	28 doNotStrip "*/armeabi-v7a/libYTCommon.so" 29 doNotStrip "*/x86/libYTCommon.so"	
∦ gradle.properties ∦ gradlew ∦ gradlew. ∦ gradlew.bat	30 doNotStrip */arm64-v8a/libYTCommon. so" 31 } 32 }	

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

Configuring Permissions

Configure permissions for your application in AndroidManifest.xml . LiteAVSDK needs the following permissions:

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



```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-feature android:name="android.hardware.Camera"/>
<uses-feature android:name="android.hardware.camera.autofocus" />
```

Configuring License

Log in to the CSS console, go to **MLVB SDK** > License Management, and click **Get License** to obtain a trial license. For detailed directions, see Applying for trial license. You will get two strings: a license URL and a decryption key.

Before you use LiteAVSDK features in your application, complete the following configurations (preferably in the application class).

```
public class MApplication extends Application {
  @Override
  public void onCreate() {
   super.onCreate();
   String licenceURL = ""; // The license URL obtained
   String licenceKey = ""; // The license key obtained
   TXLiveBase.getInstance().setLicence(this, licenceURL, licenceKey);
  }
}
```

Configuring Obfuscation Rules

In the proguard-rules.pro file, add LiteAVSDK classes to the "do not obfuscate" list.

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

Features

Last updated : 2022-09-14 17:26:49

We offer **MLVB Professional** for users outside the Chinese mainland. You need an professional license to unlock live streaming features such as publishing, playback, and basic filters (skin brightening and smoothing, etc.).

SDK and license

You can apply for a free trial license or purchase a license to use MLVB Professional.

Features

Module	Feature	Description	MLVB Professional
UI	Custom UI	Customizing UI. We provide a complete set of UI source code. You can use it directly or customize your own UI based on it.	1
Publishing	RTMP	Stream publishing by hosts from mobile phones (live showroom)	\checkmark
	Screen	Screen sharing by hosts from mobile phones (game streaming)	✓
Playback	RTMP	Playback over RTMP	\checkmark
	FLV	Playback over HTTP + FLV	\checkmark
	HLS	Playback over HLS (m3u8)	<i>√</i>
	WebRTC	LEB	<i>√</i>
Video on demand	Video on demand	Video playback on demand	\checkmark
Mic connect	Host-audience interaction	One-to-many mic connect between the host and audience	✓
	Anchor competition	One-to-one competition between hosts	\checkmark
Capturing and shooting	Aspect ratio	Multiple aspect ratios including 16:9, 4:3, and 1:1	\checkmark
	Definition	SD, HD, and FHD video; custom bitrate, frame rate, and GOP	1



Module	Feature	Description	MLVB Professional
	Capturing/Shooting control	Switching between the front and rear camera and lighting control	\checkmark
	Watermark	Watermarking videos	\checkmark
	Focus	Adjusting focal length	\checkmark
	Focus mode	Manual or auto focus	\checkmark
	Photo taking	Taking photos	\checkmark
	Background music	Selecting a local MP3 file as the background music for capturing/shooting	\checkmark
	Voice changing and reverb	Selecting a voice changing effect (e.g., little girl, middle- aged man) or reverb effect (e.g., karaoke room, hall) for capturing/shooting	✓
	Filters	Custom filters, which support strength adjusting	\checkmark
	Basic retouching	Skin brightening, skin smoothing, and rosy skin, which support strength adjusting	\checkmark
	Advanced filters	Eye enlarging, face slimming, chin slimming, chin adjustment, face shortening, and nose narrowing, which support strength adjusting	×
	Animated stickers	Identifying and reshaping facial features; adding stickers and widgets (provided as additional materials)	×
	AI-based keying	Identifying the foreground and replacing the background with other elements (provided as additional materials)	×
	Green screen keying	Removing the green parts of a video (for example, a green background) and replacing them with other elements	×

Support for RTC Publishing

Last updated : 2022-06-14 12:41:46

In order to improve publishing performance under poor network conditions, we have added support for RTC publishing in addition to the traditional RTMP protocol. This document compares the performance of publishing under different network conditions using the two protocols. The video below shows the effects of watching streams published over different methods.



Note:

For instructions on how to publish streams, please see Publishing from Camera.

Performance Under Normal and Poor Network Conditions

Test method

We simulated different network conditions at the publishing end and observed the playback effects (the streams were played over CDNs, and network conditions at the playback end were normal).

Parameter configuration

To prevent the use of different sources from affecting the results, we used V2TXLivePusher to publish the same video over RTMP and RTC.

Video parameters:

Parameter	Value
Resolution	720 x 1280
Bitrate	1800 Kbps
Frame rate	15

Comparison of performance under different network conditions

Frame rate



Note :

For a description of the network parameters, see Appendix: Network Parameters.



Stutter rate



Note :

For a description of the network parameters, see Appendix: Network Parameters.

Appendix: Network Parameters

Parameter	Description
Frame rate	Frames rendered per second
Packet loss	A packet loss rate of 50% means that for every 10 data packets sent, five fail to arrive at their destination.
Latency	A latency of 200 ms indicates that data packets are delivered by the network only 200 ms after they are sent by the SDK.
Transfer speed cap	A transfer speed cap of 800 Kbps means that 800 KB of data can be sent per second at most.
Stutter rate	Stutter occurs if the interval between the rendering of two consecutive frames exceeds 200 ms. Stutter rate is the total stuttering time divided by the total playback time.