实时互动-工业能源版

实践指南

腾讯云

【版权声明】

【商标声明】

【服务声明】

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

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

实践指南
视频流编码配置

最近更新时间：2024-02-19 17:19:11

> **说明：**
> SDK 提供多个视频编码参数配置，并支持动态自适应调节，用户可根据视频分辨率和重要性对视频流编码参数进行配置。

配置参数总览

在 streams_conf 的流配置中，可以对视频流的编码参数配置，支持以下参数：

参数	说明	可选	缺省值	备注
encode_width	期望编码宽度	是	与采集输入宽度 width 一致	SDK 会自动将视频帧变换到指定编码宽度
encode_height	期望编码高度	是	与采集输入高度 height 一致	SDK 会自动将视频帧变换到指定编码高度
min_width	最低编码宽度	是	与期望编码宽度 encode_width 一致	• SDK 会自动根据网络情况在 [min_width,width] 范围内调整编码宽度，对应编码高度也会按期望编码宽高比保持比例调整。 • 缺省会固定使用期望编码分辨率，无动态分辨率调整。
bps	期望画质编码码率（单位 kbps）	否	—	• 该值对应编码器设置编码码率的最大值。 • 注意：编码器设置码率和输出码率表现可能会出现不一致，具体表现由编码器决定。
min_bps	最低画质编码码率（单位 kbps）	是	与期望编码码率 bps一致	网络带宽高于 min_bps 时，编码器设置码率会在 [min_bps, bps] 范围内根据网络情况自适应调整
force_min	是否强制最低画质码率	是	0	• 0为非强制最低画质码率，网络带宽弱于 min_bps 时，会进一步降低码率，并通过

				onErrorEvent 回调提示网络带宽不足； • 1为强制最低画质码率，网络带宽弱于 min_bps 时，仍会以最低画质码率编码，可能会引发卡顿
fps	期望输入帧率	否	—	该值对应编码器输入的最大帧率，当实际输入帧率高于 fps 时，SDK 会自动降低到 fps 帧率输入到编码器
min_fps	最低编码帧率	是	25	• 编码器随编码码率变化会自适应在 [min_fps, fps] 范围内动态调整帧率 • 当 min_fps 配置与 fps 相同时，动态帧率功能关闭 • 对画面流畅性强要求场景，建议关闭动态帧率
major	是否为主要视频流	是	0	该参数用于决定不同类型视频流码率分配的优先级 • 0 为非主要视频流，码率分配优先级弱于主流 • 1 为主要视频流，会优先保证码率分配
codec	编码类型	否	—	• 0 为 H264 • 1 为 H265 • 2 为 AV1，仅部分芯片平台支持
encoder_type	是否硬编优先	是	1	• 0 软编优先，优先采用软编 • 1 硬编优先，优先采用芯片硬编

码率配置建议

码率配置的画质表现，跟编码器自身实现、编码类型以及使用场景等均密切相关。建议用户可根据实际使用场景以及对画质的要求，尝试调整编码码率参数，来确定出匹配自己场景的较优参数。下表是车辆 5G 远程驾驶场景下，GMSL 相机 + nvidia jetson 芯片编码器（硬编）的码率配置示意，可供参考：

编码分辨率	编码类型	建议 bps 参数值	建议 min_bps 参数值
1920x1080	H265	2500	1500
	H264	3000	1800

1280x720	H265	1500	500
	H264	2100	700
640x480	H265	420	200
	H264	500	200

主流配置建议

主流配置用于有明确视频流优先级区分的场景，例如远程驾驶场景前视画面相对其他画面可能会有更高的优先级。当网络带宽不足时，SDK 会优先保障主流画面画质，牺牲非主流画面画质，带宽极度受限情况下甚至可能会引起非主流画面暂停。建议用户可根据自身产品设计理念，对具有重要性的视频流开启主流配置。当多个视频流同时开启主流时，具有相同的优先级保障。

视频流类型	画质保障
主要视频流	• 网络带宽受限时，优先保障主流画质。 • 可以同时有多个主流，不同主流之间优先级相同。
非主要视频流	• 主流达到基本画质要求后，才会保障非主流视频画质。 • 当带宽极度受限且关闭强制最低画质（force_min）情况下，为保障主流画质，可能会引起非主流画面暂停。

动态编码参数调整

远端设备 SDK 支持动态对现场设备的编码参数进行实时调整以适应不同场景下对视频流编码的要求，如车辆运行监控场景和车辆远程接管场景下视频编码参数的切换。用户可按需求调用接口进行参数更新，更新过程中视频流不会出现中断。

> ⚠ **注意：**
> 为保障视频流画面连续，目前 SDK 暂不支持 codec 和 encoder_type 等编码类型参数调整。编码参数更新接口中待更新编码参数字段缺省时，会保持原有参数值不进行更新。

```
/*
* @name :  TRRO_fieldDeviceEncodeConfig
* @brief : 更新现场设备目标视频流编码参数
* @input : gwid 现场设备id
*          streams_id 目标视频流编号
*          encode_config 待更新的编码参数，json格式字符串，缺省字段将保持当前值不
进行更新。 下面是更新支持的编码参数示意
*  {
```

```
*        "fps": 30,
*        "encode_width": 1920,
*        "encode_height": 1080,
*        "bps": 3000,
*        "min_fps": 30,
*        "min_bps": 1800,
*        "force_min": 0,
*        "min_width": 1920
*   }
*/
extern "C" TRRO_EXPORT int TRRO_fieldDeviceEncodeConfig(const char*
gwid, int streams_id, const char* encode_config);
```

配置与密钥管理

最近更新时间：2024-08-26 15:23:42

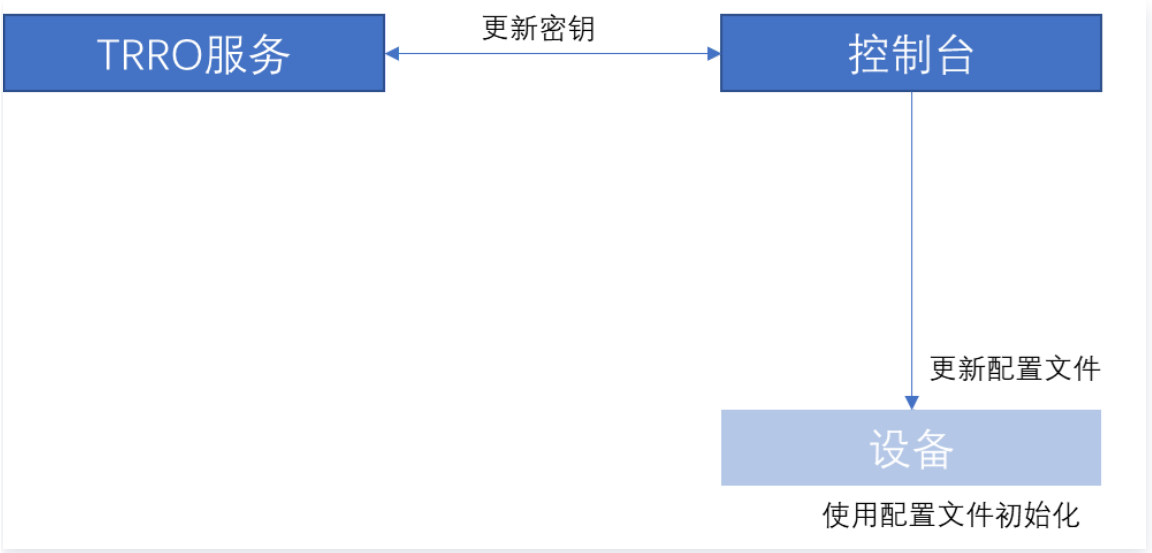
> **说明：**
> 密钥由用户生成和管理， 可自主更新并同步到 TRRO 服务侧。SDK 侧提供配置文件和 JSON 字符串两种初始化方式用于导入密钥信息。建议测试阶段可使用控制台加配置文件方式导入，商用运营阶段使用云端下发 JSON 字符串方式导入。

> **注意：**
> TRRO 服务不保存任何原始密钥消息，无法提供密钥查询服务。当密钥忘记时，请通过更新密钥方式重置密钥。

测试阶段

建议用户通过控制台的方式管理设备和密钥。

1. 通过控制台创建测试设备，并设置密钥。
2. 设备侧通过修改配置文件，更新配置文件中的密钥信息。
3. SDK 基于配置文件初始化启动。



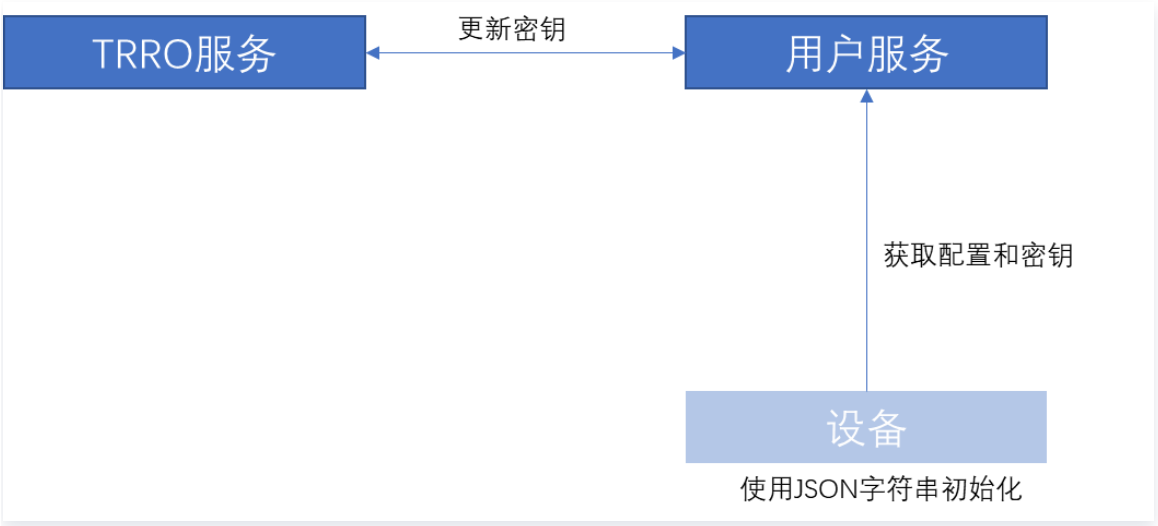
运营阶段

为保护密钥信息安全，防止被窃取，建议用户不要基于配置文件导入密钥信息。可基于 TRRO 服务提供的服务端 API 文档开发对应的用户后台服务对设备密钥进行管理，设备启动时通过安全会话连接从用户后台服务中实时获取密钥和配置 JSON 字符串，进行初始化启动。

1. 用户后台服务向 TRRO 服务注册设备，设置密钥，并按自定义策略对密钥进行更新管理。

> **注意：**
> 若希望简化设备密钥注册和更新流程，使用项目共享密钥方式替代单设备独立密钥接入，可参见 免注册登录 。

2. 设备启动时，通过安全会话连接实时向用户服务获取密钥信息和配置 JSON 字符串。
3. SDK 基于 JSON 字符串初始化启动。



设备登录管理

最近更新时间：2024-04-12 11:57:31

> **说明：**
> 设备登录是使用 TRRO 服务的前提步骤。只有完成初始化登录，SDK 才能正常工作。可通过 init 接口完成登录，通过 destroy 接口退出登录。

初始化登录

SDK 通过 init 接口完成与服务的初始化登录。

> **注意：**
> 同一时刻相同 ID 的设备仅有一个可保持登录状态，其他设备会被系统剔除登录。

SDK 提供强制登录和非强制登录两种登录方式，客户可通过配置文件或 init 接口中 JSON 字符串的 force_login 字段来指示使用何种方式进行登录。

字段	说明	缺省值	值说明
force_login	是否强制登录	0	• 0：非强制登录，若已有相同 ID 设备登录，当前设备登录会被剔除 • 1：强制登录，若已有相同 ID 设备登录，之前同 ID 的设备登录会被剔除

登录连接状态监听

SDK 提供信令状态回调接口，用于监听与信令服务的登录连接状态。

```
/**
* @name   TRRO_registerSignalStateCallback
* @brief  注册信令服务连接状态回调
* @param[in]   context                上下文
* @param[in]   callback               回调函数
* @return 1 for success, other failed
*/
extern "C" TRRO_EXPORT int TRRO_registerSignalStateCallback(void
*context, TRRO_onSignalState *callback);

/**
```

```
* @name TRRO_onSignalState
* @brief 信令连接状态回调
* @param[in]      context          上下文
* @param[in]      state            信令连接状态，参考枚举SignalState
* @return void
*/
typedef void TRRO_onSignalState(void *context, SignalState state);


/**
 * SignalState 信令连接状态枚举
 */
enum SignalState {
    kTrroReady = 0, /**< 连接建立成功 */
    kTrroLost = 1,  /**< 连接断开，内部会进行自动重连 */
    kTrroReup = 2,  /**< 自动重连成功 */
    kTrroKickout = 3,
    kTrroAuthFailed = 4,  /**< 用户名或者密码错误 */
};
```

回调状态	说明
kTrroReady	初始化登录成功。
kTrroLost	登录断开，SDK 会自动尝试重新登录。
kTrroReup	自动重连登录成功。
kTrroKickout	设备登录被剔除，剔除后 SDK 会自动 destroy 退出登录。
kTrroAuthFailed	登录认证错误。

登录退出

SDK 通过 destory 接口主动退出登录。建议不使用 SDK 阶段，可主动调用 destroy 接口，节省系统资源。

> ⚠ **注意：**
> 退出登录后，SDK 资源会被释放，接口功能无法再使用。要再次使用 SDK，可重新调用 init 接口初始化登录。

设备权限管理

最近更新时间：2024-02-20 10:59:32

> **说明：**
> 设备权限管理，用于管理远端设备和现场设备之间的会话权限。只有拥有权限的会话连接才能够被允许建立。

设备权限概念

设备权限描述的是远端设备与现场设备之间的权限配对关系。系统会基于该权限配对关系，对远端设备的列表获取、会话建立等行为进行管控。

设备配对关系	说明
远端设备与现场设备存在配对权限	• 远端设备可以看到在线登录的现场设备 • 远端设备可以向现场设备发起建立会话连接
远端设备与现场设备不存在配对权限	• 远端设备列表获取无法看到对应的现场设备 • 远端设备无法向现场设备发起建立会话连接

设备权限配置

TRRO 权限配置基于项目进行，用户可以在一个项目内部配置远端设备和现场设备的权限。支持白名单和黑名单两种配置模式。不同项目之间的设备默认是权限隔离的，无法进行会话。

> **注意：**
> 黑名单和白名单模式，在一个项目下面只能同时生效一种。可根据用法建议选用一种。

模式	说明
黑名单模式	• 名单内出现的配对：会进行权限禁用，配对间无法进行会话连接 • 名单内未出现的配对：会进行授权，配对间可以进行会话连接
白名单模式	• 名单内出现的配对：会进行授权，配对间可以进行会话连接 • 名单内未出现的配对：会进行权限禁用，配对间无法进行会话连接

可通过控制台或云 API 调用，对项目的权限名单进行配置。控制台配置方法可参见 权限管理 。云 API 调用可参看 服务端 API 文档 。

用法建议

权限配置可根据不同场景，选用不同的模式。

场景	建议模式	建议配置
无权限隔离需求（项目层面已隔离）	黑名单	无需配置
项目中有不同用户的专有设备	白名单	按专有设备的配对加入白名单中
项目中有用户共享设备	白名单	通过云 API 动态对实时共享的设备配对进行添加和删除

设备权限与控制授权关系

设备权限和控制授权（控制授权管理）是两个不同层面的权限管理。前者实现不同设备的会话安全隔离，后者解决多个授权远端设备在与同一个现场设备会话连接中的控制冲突。两者可同时使用，相互配合。

权限管理	说明
设备权限	• 管理远端设备与现场设备之间是否可以发起会话连接，解决不同设备之间会话安全隔离的问题 • 通过云 API 或控制台进行管理
控制授权	• 管理远端设备与现场设备的操控控制，解决多个远端设备与同一个现场设备会话连接中控制冲突的问题 • 需要已经具备设备权限 • 通过 SDK 接口进行管理