

# 视频外部渲染

---

## 1 常用场景

当开发者在使用 SDK 过程中出现以下任意情况时，建议使用外部渲染功能：

1. SDK 的渲染无法满足需求
2. App 使用了跨平台界面框架（例如 QT 需要有复杂层级关系的界面以实现高体验的交互）或游戏引擎（例如 Unity3D、Cocos2d-x）
3. App 需要获取 SDK 采集或拉流的视频数据进行特殊处理

## 2 下载示例源码

互动视频 Demo 源码下载地址：[互动视频示例 Demo\\_iOS](#)

[码云下载地址\(国内推荐\)](#)

[阿里云下载地址\(国内推荐\)](#)

[github下载地址](#)

示例 Demo 包含了基础专题和进阶专题，展示了如何使用 SDK API，开发者可参考其用法来实现自己的业务。示例 Demo 涵盖了 SDK 的大部分功能，开发者也可在示例 Demo 中查找 ZEGO SDK 更多的进阶功能，测试其功能和性能，以实现特定的需求。

视频外部渲染模块源码请查看 [/src/LiveRoomPlayground-iOS/ExternalVideoRenderUI](#) 目录下的源文件，该模块展示了如何渲染 BGRA32、I420 等视频码流格式的视频数据的方法。

## 3 使用步骤

外部渲染的使用流程如下：

1. 开启外部渲染
2. 初始化 SDK

3. 设置外部渲染代理对象
4. 实现外部渲染代理方法

### 3.1 开启外部渲染

调用 `ZegoExternalVideoRender` 的 `setVideoRenderType` 方法启用外部渲染，**注意：** [2020-05-12](#) 及之后版本，该接口必须在引擎启动之前调用。[2020-05-12](#) 之前版本，该接口必须在初始化 SDK 之前调用。

引擎启动的时机有：1、登录房间；2、未登录房间启动预览功能；3、未登录房间使用媒体播放器；4、未登录房间使用音效播放器。引擎关闭的时机有：1、退出房间；2、未登录房间并且预览、媒体播放器和引擎播放器都停止工作。

```
// zego-api-external-video-render-oc.h

@interface ZegoExternalVideoRender : NSObject

/**
 设置开启外部渲染时要求 SDK 提供的数据格式及渲染方式

@param type 指定 SDK 提供的数据格式及渲染方式
*/
+ (void)setVideoRenderType:(VideoRenderType)type;

@end
```

`VideoRenderType` 枚举类型如下：

枚举	说明
<code>VideoRenderTypeNone</code>	默认值，不开启外部渲染
<code>VideoRenderTypeRgb</code>	外部渲染同时 SDK 不做渲染，需 App 做渲染。外部渲染回调视频帧格式为 RGB 类别，具体类型根据操作系统和软解/硬解码器决定，iOS 正常情况下返回 BGRA32。
<code>VideoRenderTypeYuv</code>	外部渲染同时 SDK 不做渲染，需 App 做渲染。外部渲染回调视频帧格式为 YUV 类

	别，具体类型根据操作系统和软解/硬解码器决定，iOS 正常情况下返回 i420。
VideoRenderTypeAny	外部渲染同时 SDK 不做渲染，需 App 做渲染。外部渲染回调视频帧格式根据操作系统和软解/硬解码器决定，iOS 正常情况下返回 I420。
VideoRenderTypeExternalInternalRgb	外部渲染同时 SDK 也做渲染，即：可以使用 SDK 做渲染，也能获取视频源数据；且回调抛出的数据格式为 RGB 类别，iOS 默认为 BGRA32
VideoRenderTypeExternalInternalYuv	外部渲染同时 SDK 也做渲染，即：可以使用 SDK 做渲染，也能获取视频源数据；且回调抛出的数据格式为 YUV 类别，iOS 默认为 I420

#### 注意：

1. 如果选择 `VideoRenderTypeRgb`、`VideoRenderTypeYuv` 或者 `VideoRenderTypeAny` 类型，SDK 内部不会渲染推拉流视频帧数据，推流时即使调用 `setPreviewView` 设置了预览视图，调用 `startPreview` 启动本地预览时也无法显示视频画面；拉流时设置了播放视图也无法显示视频画面，需要 App 内部实现渲染逻辑。
2. 如果选择 `VideoRenderTypeExternalInternalRgb` 或者 `VideoRenderTypeExternalInternalYuv` 类型，SDK 内部渲染是有效的，即 App 设置了视图后能正常显示视频画面，但是默认不开启外部渲染，App 若想拿到视频数据需要通过 `enableVideoRender` 及 `enableVideoPreview` 开启 SDK 的外部渲染，其中 `enableVideoPreview` 开关作用是让 SDK 抛出预览的视频数据，必须在**初始化 SDK 后**调用；`enableVideoRender` 开关是让 SDK 抛出拉流的视频数据，必须在**拉流 (startPlayingStream) 后**调用。

#### 相关接口如下：

```
// zego-api-external-video-render-oc.h
```

```
@interface ZegoExternalVideoRender : NSObject
```

```
/**
```

```
* 设置是否让 SDK 抛出拉流的视频数据。必须在拉流(startPlayingStream)之后调用才生效。
```

```
* @param enable YES 开启， NO 关闭，默认为关闭
```

```

    * @param streamID 拉流的流 ID
    */
+ (BOOL)enableVideoRender:(BOOL)enable streamID:(NSString *)streamID;

/**
 * 设置是否让 SDK 抛出推流预览的视频数据。必须在 初始化 SDK 后调用。
 *
 * @param enable YES 开启, NO 关闭, 默认为关闭
 * @param channelIndex 推流通道, 默认为主通道
 */
+ (BOOL)enableVideoPreview:(BOOL)enable channelIndex:(ZegoAPIPublishChannelIndex)channelIndex;

@end

```

### 3.2 设置代理对象

通过 `ZegoExternalVideoRender` 的 `setZegoVideoRenderDelegate` 方法设置外部渲染代理对象。

```

// zego-api-external-video-render-oc.h

@interface ZegoExternalVideoRender : NSObject

/**
 获取ZegoExternalVideoRender 单例

  @return ZegoExternalVideoRender 单例对象
  */
+ (instancetype)sharedInstance;

/**
 设置外部渲染回调

  @param delegate 外部渲染回调代理, 用于接收待渲染的视频数据
  */
- (void)setZegoVideoRenderDelegate:(id<ZegoVideoRenderDelegate>)delegate;

@end

```

### 3.3 实现代理方法

请注意：若需要渲染预览视图，需在推流前先调用 `startPreview` 启动预览，否则收不到 SDK 的视频数据回调。

### 3.3.1 实现外部渲染回调接口

```
// zego-api-external-video-render-oc.h

/**
 SDK 待渲染视频数据

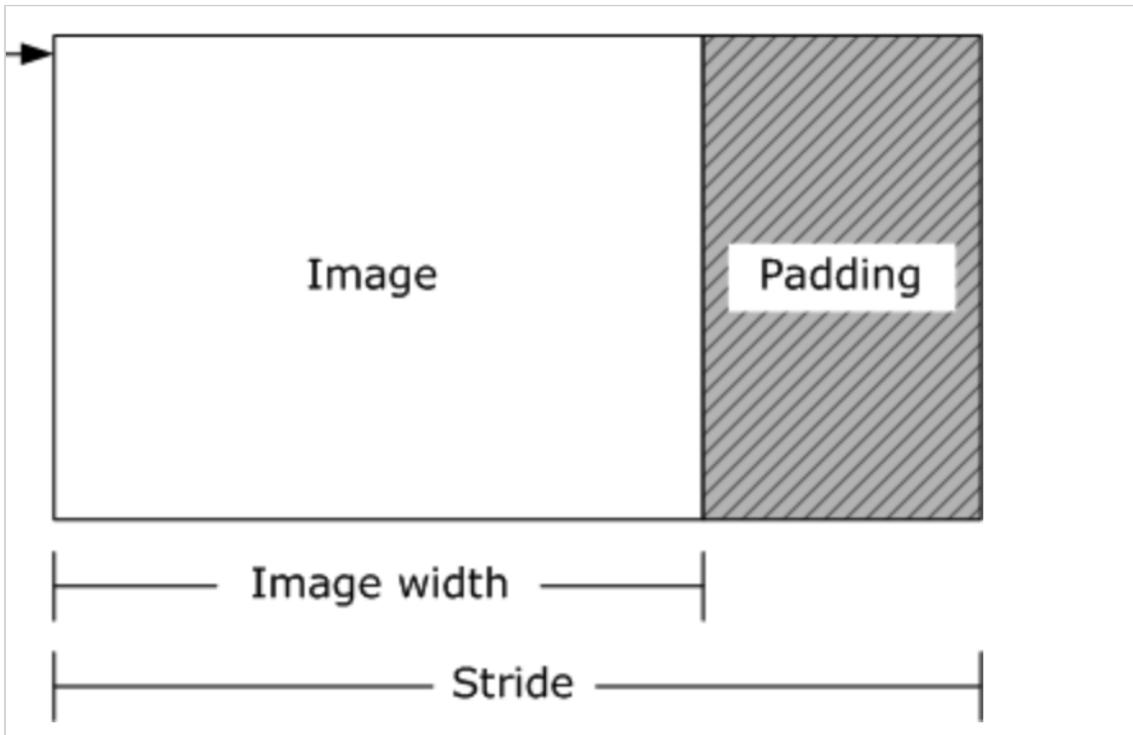
 @param data 待渲染数据, 当 VideoRenderType 设置为 VideoRenderTypeExternalInternalRgb
 或者 VideoRenderTypeExternalInternalYuv 时, SDK 会使用修改后的 data 进行渲染
 @param dataLen 待渲染数据每个平面的数据大小, 共 4 个面
 @param width 图像宽
 @param height 图像高
 @param strides 每个平面一行字节数, 共 4 个面 (RGBA 只需考虑 strides[0])
 @param pixelFormat format type, 用于指定 data 的数据类型
 @streamID 流名
 */
- (void)onVideoRenderCallback:(unsigned char **)data dataLen:(int*)dataLen width:(int)
width height:(int)height strides:(int[])strides pixelFormat:(VideoPixelFormat)pixelFor
mat streamID:(const char *)streamID;

/**
 SDK 通知下一帧数据是否需要翻转

 @param mode 翻转类型
 @param streamID 流名
 @discussion 仅本地预览的外部渲染会回调。此处的 mode 是基于推流图像计算出来的, 和 SetV
ideoMirrorMode 不一定一致, 请基于 SetFlipMode 的参数决定是否翻转
 */
- (void)onSetFlipMode:(int)mode streamID:(NSString *)streamID;
```

请注意： 当使用1) 拉流, 2) VideoRenderType 设置为 VideoRenderTypeExternalInternalRgb 或者 VideoRenderTypeExternalInternalYuv 时, 用户可以修改回调的 data, SDK 会使用修改后的 data 进行渲染。

strides和图像之间的关系如图：



### 3.4 视频帧格式

视频帧格式枚举	说明
PixelFormatUnknown	UNKNOWN
PixelFormatI420	I420 (Y, U, V 3个平面, 各自平面存相应数据)
PixelFormatNV12	NV12 (Y, UV 2个平面, UV平面数据以UV UV顺序排列)
PixelFormatNV21	NV21 (Y, VU 2个平面, VU平面数据以VUVU顺序排列)
PixelFormatBGRA32	BGRA32
PixelFormatRGBA32	RGBA32
PixelFormatARGB32	ARGB32
PixelFormatABGR32	ABGR32
PixelFormatI422	I422

至此，App 成功获得 SDK 采集的视频数据，用于实际的渲染动作或者进行深加工操作。

请注意，开发者需要按照各自的需求特点，对 SDK 拷贝给 App 的视频数据做处理。

## 4 Q&A

- Q1: 视频外部渲染，分为推流外部渲染、拉流外部渲染，推流和拉流外部渲染分别有什么作用？什么场景下使用？
- A1: 推流外部渲染是让主播看到额外的渲染效果，在原视频画面基础上加上特效等；拉流外部渲染是让观众看到不同的渲染画面，可以根据观众的喜好，在推流画面的基础上加上特效。
- Q2: 如果使用外部渲染 `VideoRenderTypeRgb`、`VideoRenderTypeYuv` 或者 `VideoRenderTypeAny` 类型时，预览及拉流接口中的 `view` 参数填什么？
- A2: `view` 在该渲染模式中，SDK 的内部渲染无效，`view` 因此预览时可以不调用设置预览视图接口或置空，设置拉流视图可置空。
- Q3: 推流时使用视频外部渲染功能，处理过的视频数据buffer只会在本地展示，不会加入到推流当中吧？
- A3: 只会在本地展示，不影响推流出去的视频数据。
- Q4: 外部渲染的视频帧的宽高是多少？
- A4: 外部渲染的视频帧的宽高会在代理方法中返回，其值与推流时设置的分辨率相同，比如推流默认的是  $540 * 960$ ，回调返回的宽高就是  $540 * 960$ 。
- Q5: 外部渲染出来的视频格式是什么？是否支持 yuv？
- A5: 根据选择的渲染类型和软解/硬解码后直接返回的数据决定，查看[视频帧格式枚举]；支持 yuv，例如 `PixelFormatNV12`，`PixelFormatNV21`，`PixelFormatI420`。
- Q6: 外部渲染每秒回调的频率是多少？有什么需要注意的吗？

- A6: 推流外部渲染回调的频率一般是和推流时设置的帧率相同，但是如果开启了流量控制并且控制属性中包含帧率的话，推流外部渲染的回调频率是会随之改变的；而拉流外部渲染的回调频率也是会随着接收到的视频数据帧率改变的，比如推流端开启流量控制导致帧率改变、拉流端网络卡顿、拉流端网络恢复 SDK 开始追帧，都会影响拉流外部渲染的调用频率。
- Q7: 外部渲染，如何拿到第一帧数据呢？
- A7: `ZegoVideoRenderDelegate` 的方法 `onVideoRenderCallback:dataLen:width:height:strides:pixelFormat:streamID:` 回调第一次返回的数据。
- Q8: 是否可以推流渲染我们自己做，拉流渲染由 ZEGO SDK 做？
- A8: 支持，使用 `VideoRenderTypeExternalInternalRgb` 或 `VideoRenderTypeExternalInternalYuv` 渲染类型，调用 `enableVideoPreview` 开启外部渲染，拿到 SDK 抛出的视频数据后自行渲染，此接口默认 SDK 会做渲染。
- Q9: 渲染预览视图时，为什么收不到视频数据的回调？
- A9: 如果需要渲染预览视图，需要在推流前先启动预览（`startPreview`），否则收不到视频数据的回调。