本文介绍如何使用三体 windows SDK 快速实现音视频通话。

Demo 体验

三体在 Github 上提供开源的实时视频通话 Demo 项目 3TLiveDemo。在开始对接三体 SDK 之前，您可以通过该示例体验实时音视频通话效果。

开发环境:

- 支持 Microsoft Visual Studio 2015 或者以上版本
- 支持 Windows 7 及以上版本的操作系统

创建 Windows 项目

参考下列步骤创建一个 windows 项目；如果已有 Windows 项目，请直接参考集成 SDK

- 打开 Microsoft Visual Studio 并点击新建项目。
- 进入新建项目窗口，选择项目类型为 MFC 应用程序，输入项目名称，选择项目存储路径，并点击确认。
- 进入 MFC 应用程序窗口，选择应用程序类型为基于对话框，并点击完成。

集成 SDK

1、拷贝 SDK 文件

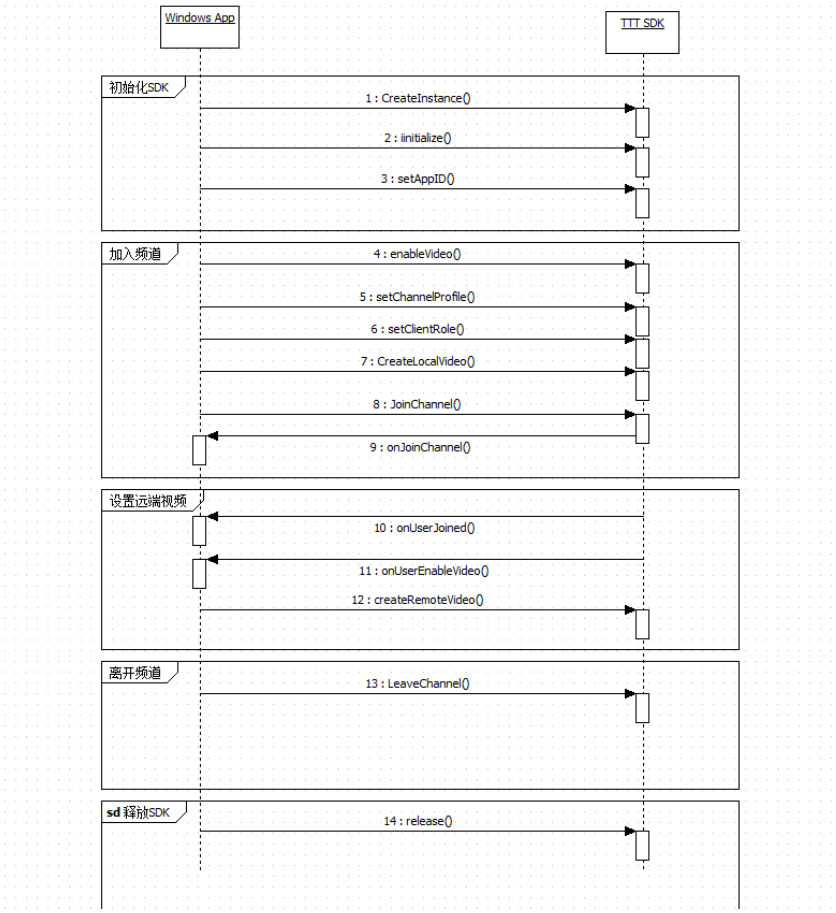
- 从官网上 SDK 下载处下载最新的 SDK 包，解压
- 将解开的文件目录复制到 Windows 项目文件目录下。

2、配置工程属性

- 进入 **C/C++ > 常规 > 附加包含目录**菜单，点击**编辑**，并在弹出窗口中输入 **$(SolutionDir)include**。
- 进入**链接器 > 常规 > 附加库目录**菜单，点击**编辑**，并在弹出窗口中输入 **$(SolutionDir)lib**。
- 进入**链接器 > 输入 > 附加依赖项**菜单，点击**编辑**，并在弹出窗口中输入 TTT**RtcSDK.lib**。

实时音视频通话的建立

视频通话的时序图如下：



1、 初始化 SDK

在调用其他 SDK 接口之前，需要创建并初始化 IRtcEngine 对象。

- CreateInstance()，创建 IRtcEngine 对象
- Initialize(RtcEngineContext)，初始化 IRtcEngine 对象；设置 IRtcEngine 对象的事件回调方法，以及日志输出路径和级别。

- setAppID()，设置应用 AppID。AppID 可以在官网上免费注册获取。

回调接口定义：

```
#include "TTTRtcEngine.h"
#include "TTTstruct.h"

using namespace TTTRtc;

class C3TEngineEventHandler : public IRtcEngineEventHandler
{
	public:
	C3TEngineEventHandler();
	virtual ~C3TEngineEventHandler();

	//加入频道的回调
```

```
        virtual void onJoinChannel(int64_t channel, int64_t userID, RtcErrorCode result);
        virtual void onError(int err, const char* msg);
        virtual void onWarning(int warn, const char* msg);
        //离开频道的回调
        virtual void onLeaveChannel(RtcErrorCode reason);
        //其他用户进入频道的回调
        virtual void onUserJoined(int64_t userID, CLIENT_ROLE_TYPE role);
        //其他用户离开频道的回调
        virtual void onUserOffline(int64_t userID, RtcErrorCode reason);
        //其他用户视频打开/关闭的回调
        virtual void onUserEnableVideo(int64_t userID, const char *mediaID, int mediaType, bool enabled) override;
};

//初始化IRtcEngine
RtcEngineContext context;
g_TTTEngine = IRtcEngine::createInstance();

if (g_TTTEngine != NULL)
{
        //设置回调方法
        context.eventHandler = &g_3TEngineEventHandler;

        int res = g_TTTEngine->initialize(context);
        if (res == 0)
        {
                //初始化成功，设置appID
                g_TTTEngine->setAppID(g_LocalUser.m_sAppID.c_str());
        }
}
```

2、 加入频道

- enableVideo()，允许使用视频功能
- setChannelProfile()，设置频道模式，支持直播模式和通讯模式。两种模式的不同请参考官网上 API 接口的详细描述。
- setClientRole()，设置用户角色，直播模式支持主播、副播和观众三种角色，每一个频道只能有一个主播，默认情况下，只有主播进了房间，其他角色的用户才能进房间；通讯模式只支持副播角色
- CreateLocalVideo()，设置本地视频。本地视频支持摄像头采集、屏幕分享、以及外部视频源三种。本地视频显示需要调用 StartPreview()
- 完成以上参数设置后，我们就可以调用 JoinChannel 加入频道。加入频道需要输入频道号、token 和用户 ID。

  频道号是一个 int64_t 的整数，用户可以自己设定自己的频道号。

用户 ID 是一个 int64_t 的整数，用户可以自己设定用户 ID，同一个频道内的用户 ID 不能重复。

Token，用户安全识别码，默认情况下使用 NULL，不进行身份认证识别；用户可以通过我们的后台申请提高安全级别，具体使用详见：生成 token

onJoinChannel 返回加入频道的状态信息。在这里返回成功或者失败的信息。

示例代码：

```
int CDialog2::joinChannel()
{
	if (g_TTTEngine != NULL)
	{
		g_TTTEngine->enableVideo();
		g_TTTEngine->setChannelProfile(CHANNEL_PROFILE_LIVE_BROADCASTING);
		g_TTTEngine->setClientRole(CLIENT_ROLE_BROADCASTER, NULL);

		LocalVideoConfig cfg2;
		cfg2.userID = g_LocalUser.m_uid;
		cfg2.type = VIDEO_CAPTURE_CAMERA;
		cfg2.devIndex= 0;
		cfg2.screenRect.x = 0;
		cfg2.screenRect.y = 0;
		cfg2.screenRect.w = GetSystemMetrics(SM_CXSCREEN);
		cfg2.screenRect.h = GetSystemMetrics(SM_CYSCREEN);
		cfg2.width = 1280;
		cfg2.height = 720;
		cfg2.framerate = 15;

		CStatic	*pStatic = NULL;
		pStatic = (CStatic*)GetDlgItem(IDC_STATIC_VIDEO1);
		cfg2.viewHwnd = pStatic->GetSafeHwnd();

		char* id = g_TTTEngine->createLocalVideo(cfg2);
		g_TTTEngine->joinChannel(123123, "", 1001);
	}
	return 0;
}
```

3、设置远端视频

- onUserJoined()，表示有一个远端用户加入房间。在这个回调里获得远端用户的 ID 和角色类型
- onUserEnableVideo()，表示一个远端用户上传或者关闭了一路视频。回调 mediaID 是唯一标识一个视频流的 ID；enabled 为 true 时表示该视频流已经上行，可以打开本路视频，为 false 时表示视频已经关闭，需要释放本路视频。

- CreateRemoteVideo()，mediaID 表示需要打开的视频流 ID；hwnd 表示视频流显示的窗体句柄；renderScaleType 表示渲染模式，包含补黑边和充满窗体两种模式。

主副播分别实现到这一步，一个基本的一对一通话就完成了。

```
void onUserEnableVideo(int64_t userID, const char *mediaID, int mediaType, bool enabled)
{
	RemoteVideoConfig cfg ;
	cfg->renderScaleType = VIDEO_SCALE_FIT;
	uint64_t uid = userID;
	std::string mid = mediaID;
	if (true == enabled)
	{
		CStatic	*pStatic = NULL;
		pStatic = (CStatic*)GetDlgItem(IDC_STATIC_MAINVIDEO);
		cfg->hwnd = pStatic->GetSafeHwnd();
		if (g_TTTEngine != NULL)
		{
			int rs = g_TTTEngine->createRemoteVideo(*cfg);
		}
	}
	else
	{
		if (g_TTTEngine)
			g_TTTEngine->releaseRemoteVideo(cfg->userID, mid.c_str());
	}
	return 0;
}
```

4、 离开频道

结束通话时，需要调用 LeaveChannel 来离开当前通话。

5、 释放 SDK

Release（）释放 SDK 资源。

示例代码

3TLiveDemo