代码结构

代码根目录为HEVC\_Desktop\_Streaming\x265\_1.7\source。

通用代码包括：

* common\network\common\_utils.hpp 定义用到的数据结构如SharedBuffer和全局函数。
* common\network\connection.hpp 网络连接类TcpConnection，MsgConnection，DataConnection的定义和实现。

Server 端代码包括：

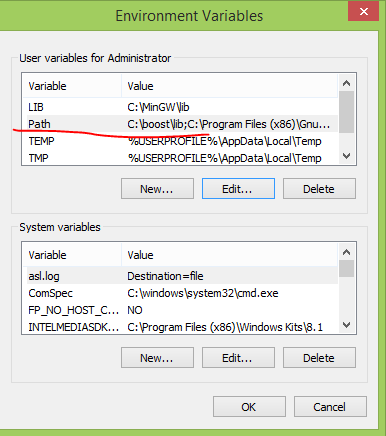
* Common\network\service.hpp(cpp) 网络服务类的定义和实现
* Common\network\desktop\_streaming\_service.hpp 本系统主类定义及部分实现
* Input\yuv.cpp 对原x265读入模块进行了改造，使之从 capture frame buffer 中读取yuv帧。
* Input\capture.hpp 桌面捕捉功能实现，暂用GDI
* Output\raw.cpp 对原x265输出模块进行改造，将编好的帧发送到网络上。
* X265.cpp 原x265主程序，现在有DesktopStreamingService的DoStartEncoder启动x265\_main实现编码。
* server.cpp server主程序

Client 端代码包括：

* Common\network\request.hpp 网络服务请求类的定义和实现。
* Common\network\desktop\_streaming\_request.hpp(cpp) DesktopStreaming 请求的定义和实现。播放器和存jpg文件功能在PlayerRoutine函数中。
* Common\network\client.cpp client主程序

编译库依赖

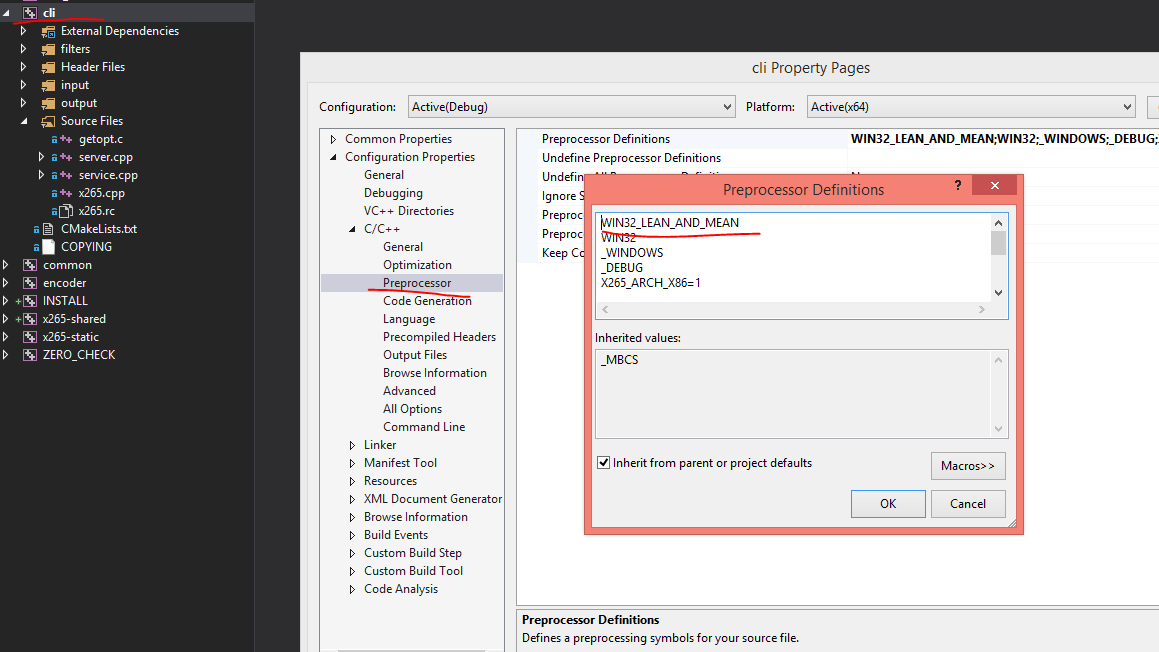
网络通信模块用boost asio实现，需要安装boost库的头文件和lib文件（本机已安装到C:\boost目录下），并将boost lib目录添加到path环境变量中。



在vs工程属性中设置相应的include文件夹和library dependency.

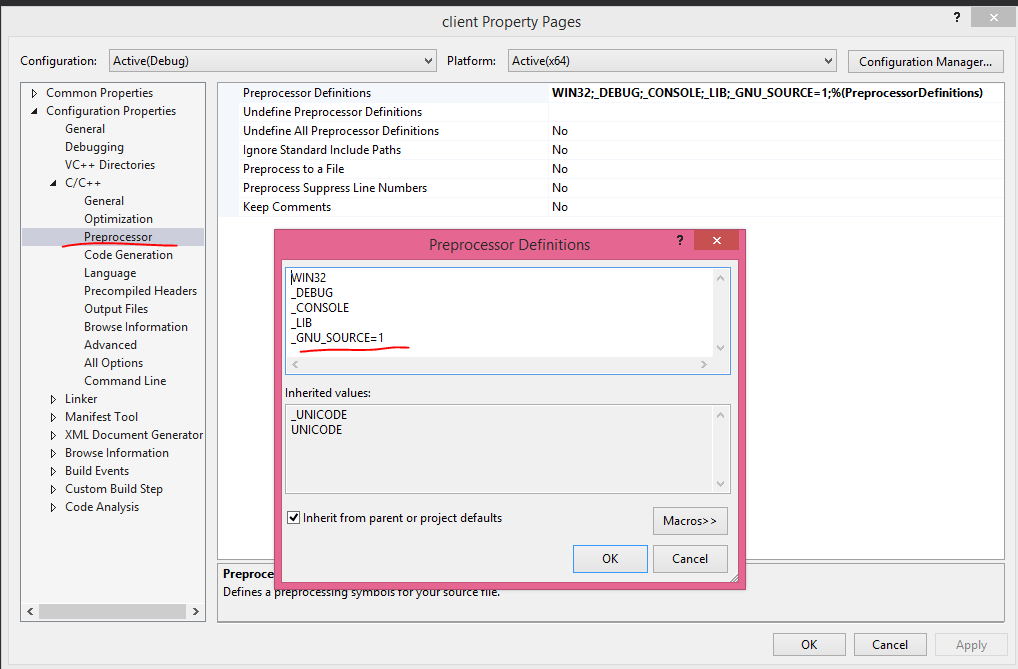
Server 端工程其他配置就是x265的配置，Sln文件在HEVC\_Desktop\_Streaming\x265\_1.7\build\vc12-x86\_64\x265.sln

在使用boost asio之后编译会提示winsock2.h 已经被导入过的编译错误，解决方法是在cli工程属性中添加WIN32\_LEAN\_AND\_MEAN宏定义。



Client 端需要用ffmpeg解码，用SDL绘制播放器并显示视频。编译依赖的头文件和库文件在client的工程目录HEVC\_Desktop\_Streaming\client\vs\_test2下的include和lib目录。其中lib目录仅存放了编译时用到的.lib文件，如avcodec.lib SDL.lib，运行时用到的dll文件avcodec-56.dll SDL.dll 需要放在path环境变量指定的目录中，为了方便在本机上和boost运行时库放在一起C:\boost\lib。

根据SDL的编译要求，需要定义宏\_GNU\_SOURCE=1



运行说明

本系统目前仅支持1920x1080 32bit 真彩色桌面，相应参数被hard coded at capture.hpp:271

static YuvFrame frame(1920, 1080, X265\_CSP\_I444);

x265 编码所用到的参数 hard coded at desktop\_streaming\_request.hpp:56

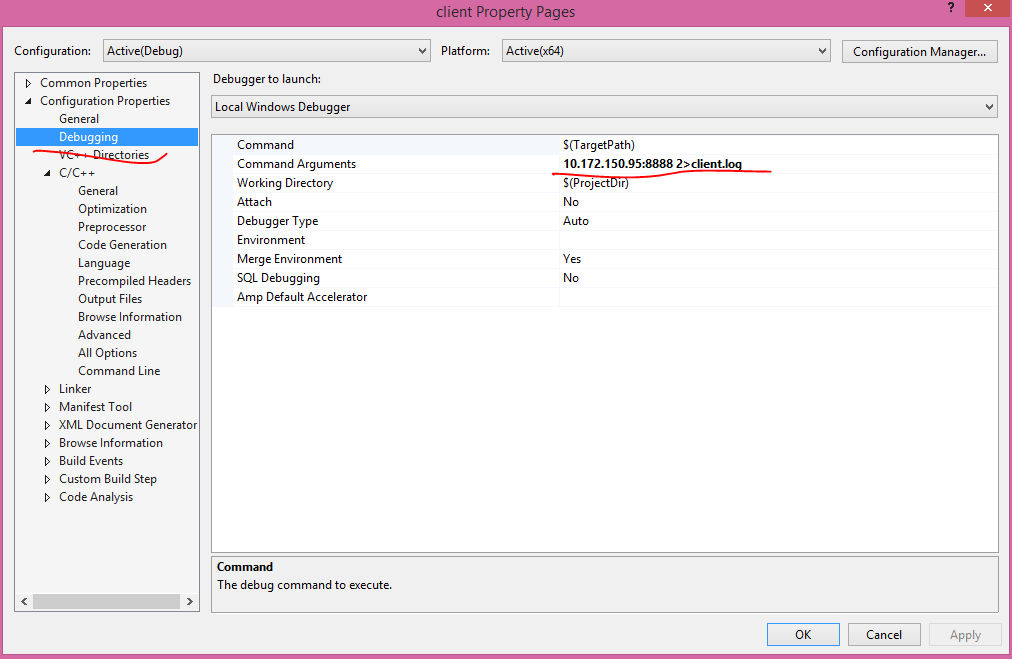
StringPtr pMsg = std::make\_shared<std::string>("x265 - --preset ultrafast --bframes 0 --rc-lookahead 0 --ref 1 --no-b-pyramid --input-res 1920x1080 --input-csp i444 --fps 60 -o -\n");

其中x265 后面的”-” 愿意从标准输入读入原始yuv数据，”-o -” 愿意将编码后的帧写到标准输出上。由于对input和output模块都做过改写，这两个参数在这里无实际意义。

## 启动

Server的启动不需要其他参数，若想将log写入到文件，则在命令行后加入 2>server.log (文件名任意，不一定是server.log)

Client的启动格式为 .\client $serverip:8888（服务端口定为8888用于client server命令交互）为了能和server做后续交互，建议将log输出重定向到文件中，否则log输出会不断刷命令行窗口。.\client 10.172.150.95:8888。在client工程属性中已经做了设置：



Server地址变更后应相应更新这个设置。

## 命令控制

Streaming开始后，可以在client命令行输入如下命令控制streaming

* “pause”: 暂停不间断的capture。
* 直接输入数字n：（需要在暂停之后）capture并encode发送下n帧。
* “start”: 恢复不间断的capture。
* “quit”: 退出streaming系统。