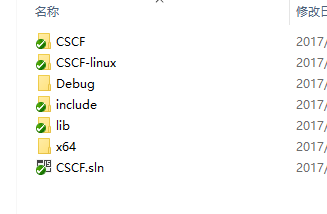
# Vowifi服务器端搭建文档

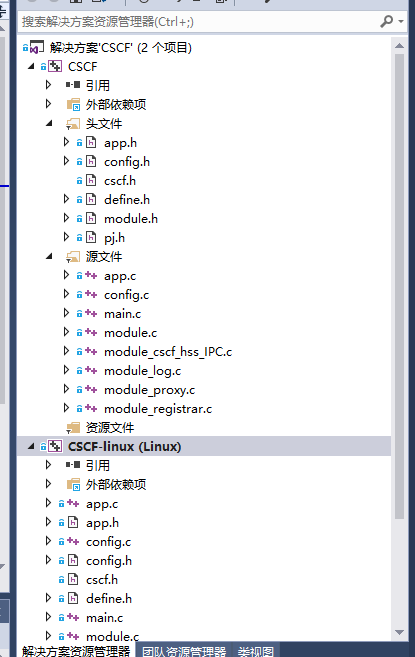
## 文件结构说明

本项目的文件结构如下图所示：



* + CSCF.sln：项目的解决方案文件；
  + CSCF：项目的所有源代码与头文件；
  + CSCF-linux：VS远程调试的相关配置文件，由VS生成；
  + Include：pjsip库的头文件；
  + Lib：pjsip库编译得到的lib文件（Windows平台，x64）与a文件（CentOS7，x64）。

## 项目结构说明



项目结构如上图所示，CSCF.sln解决方案中共有两个项目，CSCF为Windows调试时的启动项目，在Windows环境中可以通过该项目进行断点、单步等调试，配合VS可以极大的降低调试难度；CSCF-linux为Linux调试时的启动项目，VS会自动拷贝所有源代码至远程服务器并通过GDB远程调试。

需要注意的是，两个项目共用一份源码，仅仅是启动方式不同，修改代码时可以在任意项目下修改。

## 代码结构说明

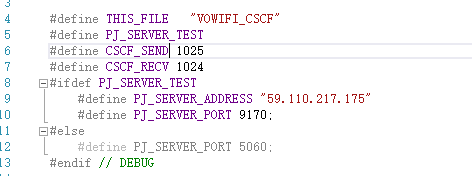
各个源代码文件的简要说明如下：

* + pj.h：包括了pjsip库的头文件，以及windows下对lib文件的处理，需要被每个源文件包含；
  + cscf.h：包括了本项目中需要使用的头文件，需要被每个源文件包含；
  + define.h：项目的宏定义均在此头文件中，远程调试时需要对其进行配置；
  + app.c：程序的初始化、销毁函数以及pjsip的工作进程；
  + app.h：声明并定义pjsip中各个基本对象，如endpoint、线程池等；
  + config.c、config.h：程序运行时的相关设置；
  + module.c：模块的注册函数，添加新模块时需要在其中注册；
  + module.h：各个模块的声明，包括模块本身、模块的处理函数、以及模块需要的数据结构，需要被每个模块的实现文件包含；
  + 以module\_开头的c文件：模块的实现文件。

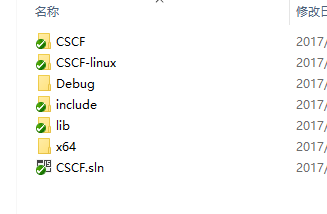
## 开发环境搭建指南

本项目使用Visual Studio 2017开发，支持Windows和Linux两个平台下的运行与调试。Windows下的调试环境已经通过在项目中配置好，无需进一步设置。Linux中可以按照以下步骤搭建开发环境：

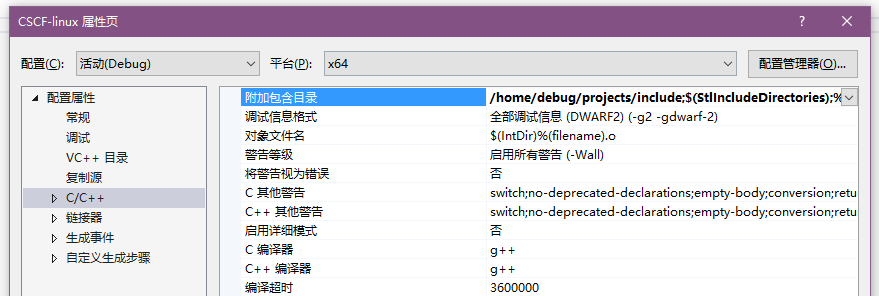
1. Clone项目至本地目录；
2. 运行CSCF\CSCF.sln，开启解决方案；
3. 在VS的选项中选择跨平台，并填入远程主机的IP地址、用户名以及密码；
4. 在define.h中进行以下配置：



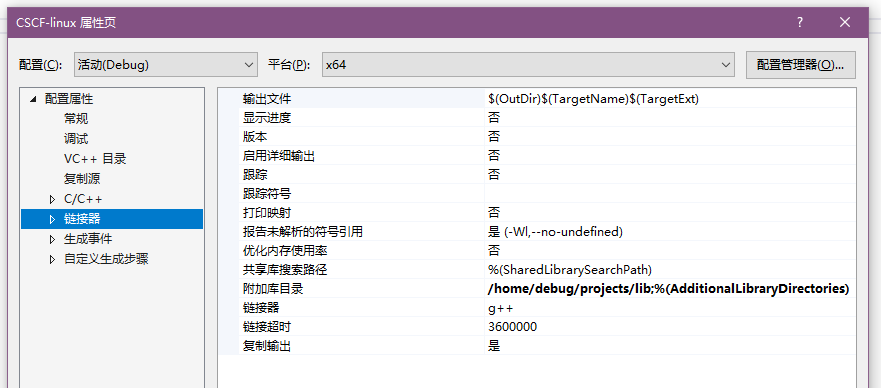
1. 若需要在远程Linux主机中调试，请定义PJ\_SERVER\_TEST宏，并修改PJ\_SERVER\_ADDRESS为远程服务器的外网地址，修改PJ\_SERVER\_PORT为远程服务器收发SIP信令的端口；
2. 若需要在本地环境调试，请不要定义PJ\_SERVER\_TEST宏，此模式下没有HSS路由表，仅用于调试；
3. CSCF\_SEND、CSCF\_RECV为CSCF与HSS通信的消息队列的键，可以根据需要修改。
4. 在远端服务器进行以下配置：



1. 为了VS中Linux调试工具的正常使用，需要在远端服务器中安装openssh-server、g++、gdb与gdbserver；
2. 将项目目录中的include文件夹拷贝至远端服务器，并将其**绝对路径**填入CSCF-linux项目属性中“C/C++选项”中的“附加包含目录”中，如下图所示；



1. 将项目目录中的lib文件夹拷贝至远端服务器，并将其**绝对路径**填入CSCF-linux 项目属性中“链接器”选项中的“附加库目录”中，如下图所示（注：lib中的库文件是在CentOS7 x64环境下编译的，如果运行在其他Linux系统或运行在x86环境下，则需要通过pjsip重新编译库文件）；



1. 可以通过VS 调试该项目，也可以在Linux中直接运行该项目。VS会将.out文件生成在 **用户目录/projects/CSCF-linux/bin/x64/Debug**中。