# Win10 Visual Studio 2022配置安装jsoncpp

# jsoncpp安装：

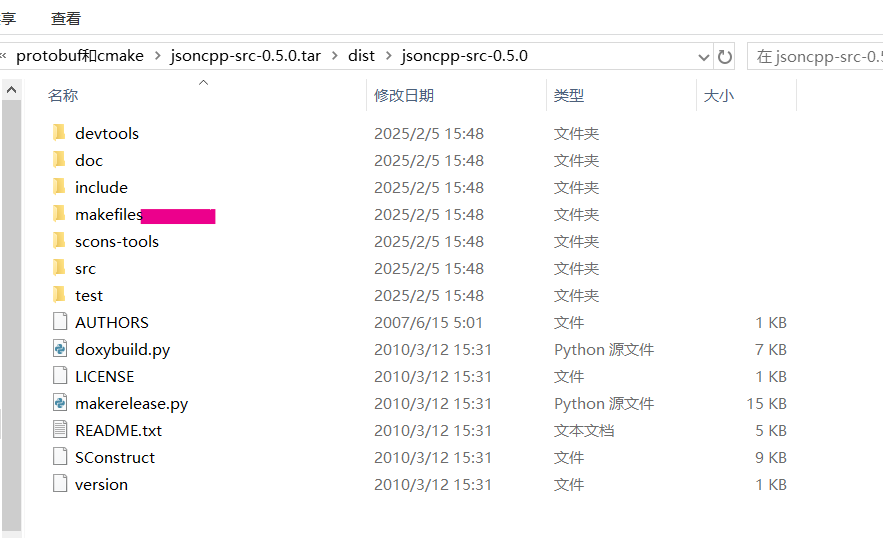
jsoncpp下载地址： <https://github.com/open-source-parsers/jsoncpp> 官方文档： <http://jsoncpp.sourceforge.net/old.html>

选择windows版本的下载。如果下载速度很慢，可以去llfc的网盘地址下载 <https://pan.baidu.com/s/1Yg9Usdc3T-CYhyr9GiePCw?pwd=ng6x>

验证码ng6x

# jsoncpp环境搭建：

下载后解压文件夹：



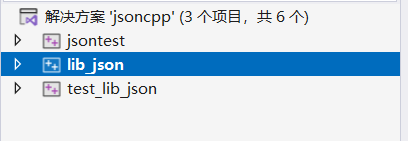
然后，进入makefiles文件夹下的vs71文件夹：



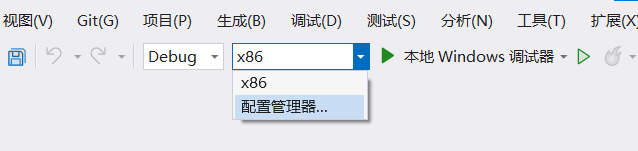
找到jsoncpp.sln文件，用visual studio打开，因为我的是visual studio2022版本，所以会提示我升级：



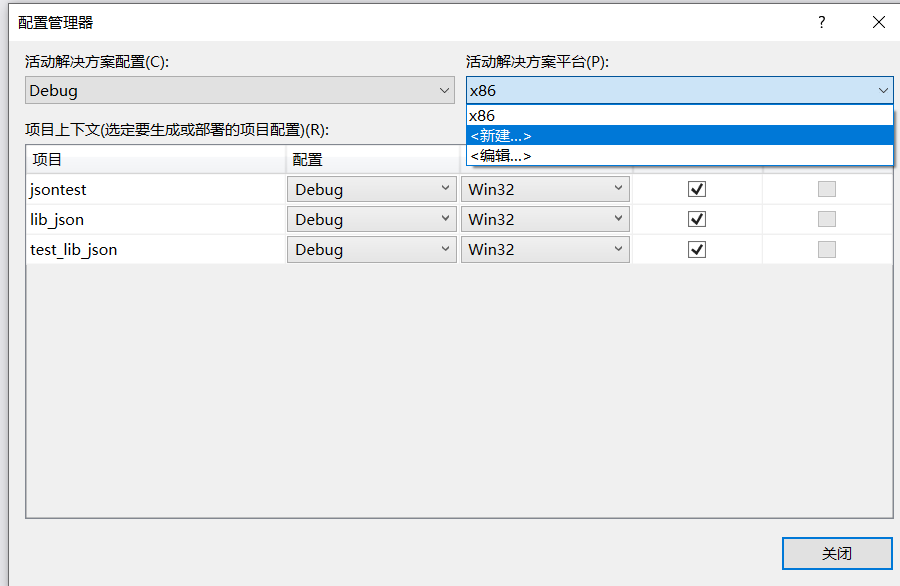
点击确定，然后编译右边的lib\_json库即可，其余两个都是测试的：



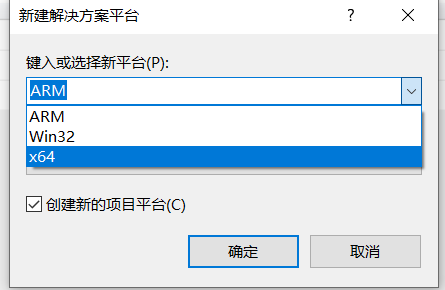
配置编译属性，我想要用x64位的，所以配置编译平台为X64位，在这选择配置管理器：



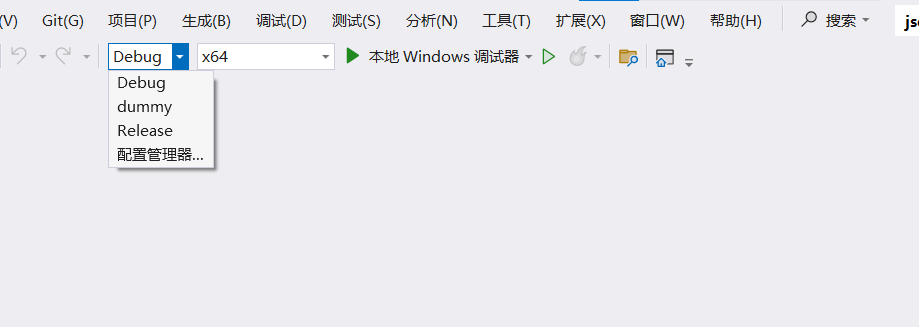
选择新建：



选择x64点击确定：

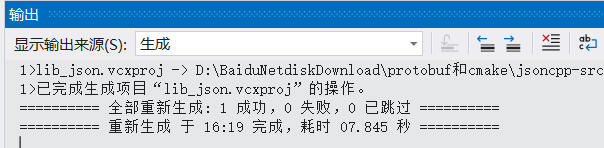


在这里选择编译debug还是release，编译模式为debug模式，最好把release版和debug版都编译一遍：

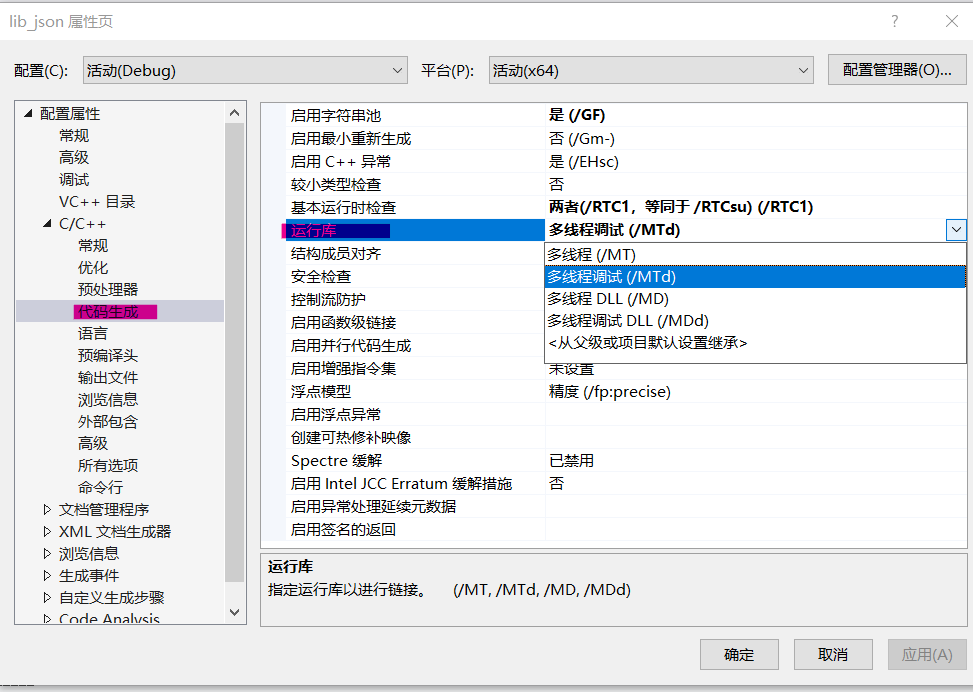


右键lib\_json项目，点击重新生成：





有时生成的和其他库不兼容，是因为在C/C++目录下的代码生成目录不相同，可以右键项目属性更改：



如果之前在配置boost时使用了命令：

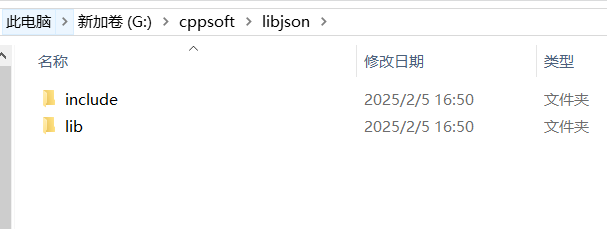
.\b2.exe install --toolset=msvc-14.2 --build-type=complete --prefix="D:\cppsoft\boost\_1\_81\_0" link=static runtime-link=shared threading=multi debug release

上面命令的shared就相当于“多线程调试DLL(/MDd)“那么我们debug要选择“多线程调试DLL(/MDd)”与之匹配，release要选择多线程DLL(/MD)。

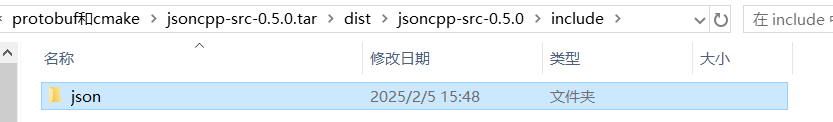
编译成功之后，就会在vs71文件夹下的Debug/release文件夹下生成：

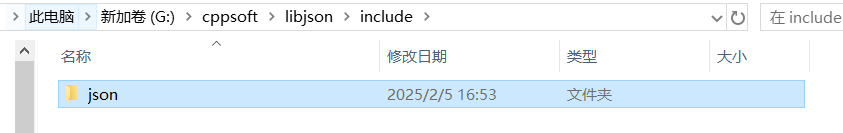


在自己要放置软件环境的地方新建一个libjson文件夹，新建一个include文件夹和一个lib文件夹：



把原先解压的include文件夹下的json拷贝到自己新建的include中：



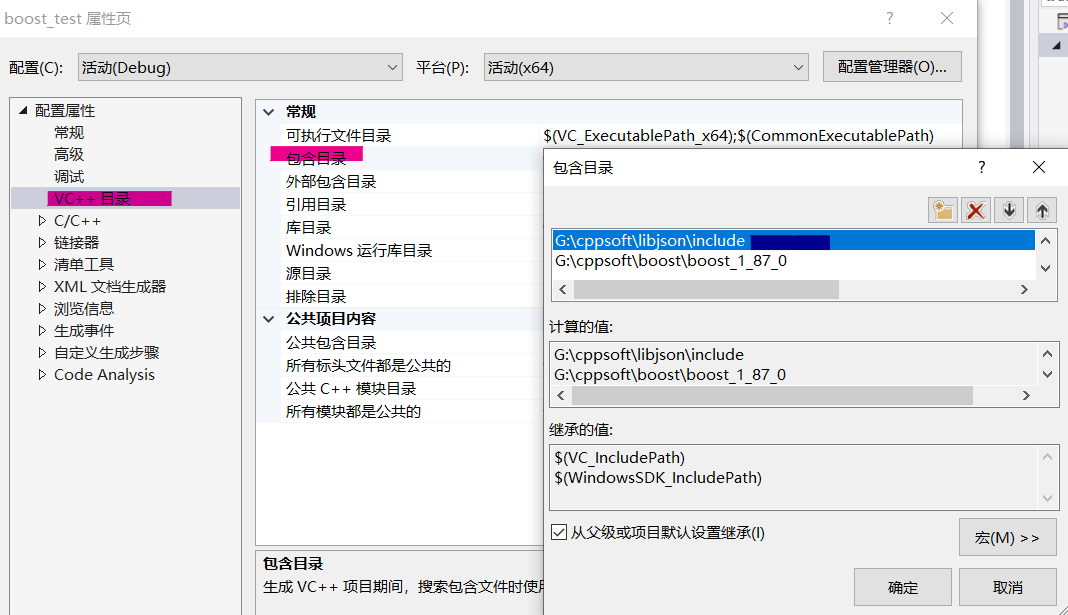


lib文件夹中放刚才生成的两个.lib文件：

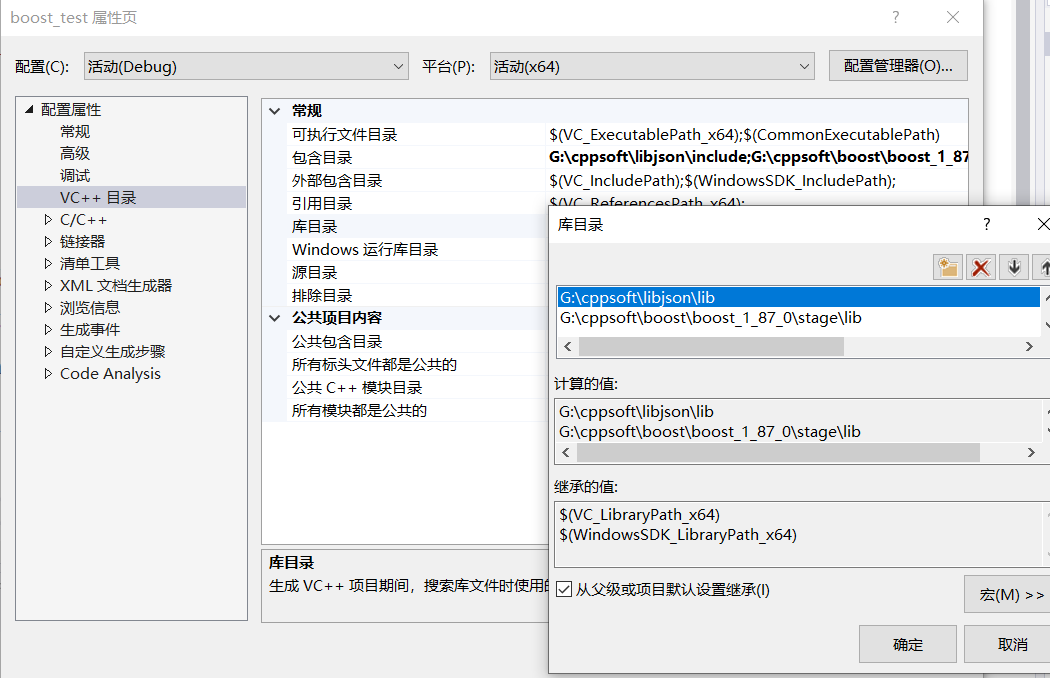


# jsoncpp配置到项目中：

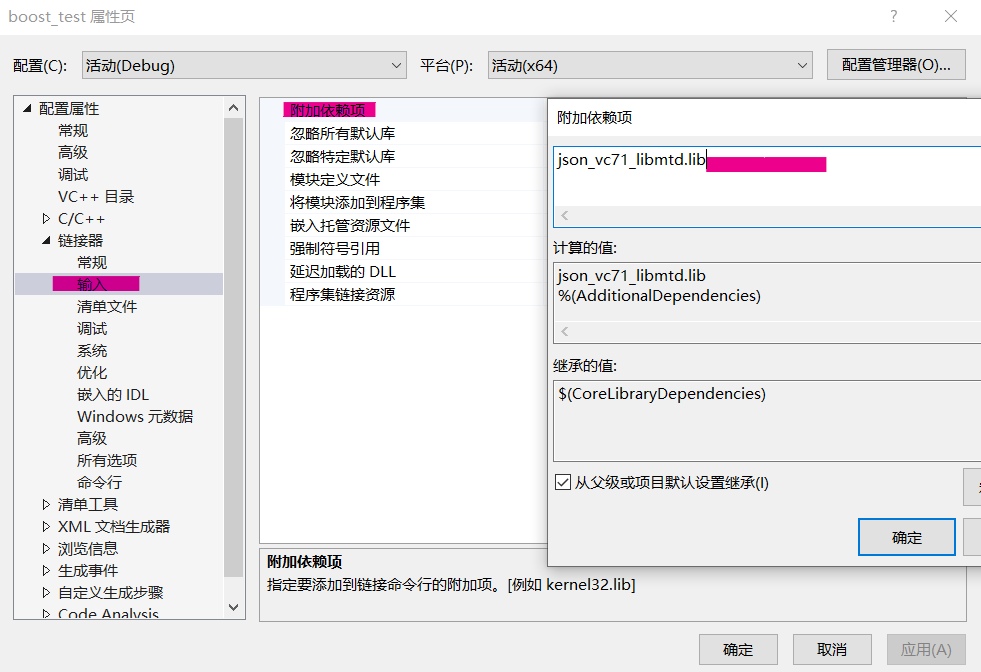
新建项目，右键属性，选择VC++目录，选择包含目录，包含到自己之前创建的的include文件夹下：



库目录包含到之前创建的lib文件夹下：



点击连接器选项的输入选项，用到哪个库，就把那个库的名字输入到附加依赖项中：



点击应用确定即可。

# 测试jsoncpp是否可用：

使用代码测试是否可用：

#include <iostream>

#include <json/json.h>

#include <json/value.h>

#include <json/reader.h>

int main()

{

Json::Value root;

root["id"] = 1001;

root["data"] = "hello world";

std::string request = root.toStyledString();

std::cout << "request is " << request << std::endl;

Json::Value root2;

Json::Reader reader;

reader.parse(request, root2);

std::cout << "msg id is " << root2["id"] << " msg is " << root2["data"] << std::endl;

}

运行结果如下即为成功：

