# 一、Log4cPlus简介

log4cplus是C++编写的开源的日志系统，具有线程安全、灵活、以及多粒度控制的特点，通过将日志划分优先级使其可以面向程序调试、运行、测试、和维护等生命周期。用户可以选择将日志输出到屏幕、文件、NT event log、甚至是远程服务器，通过指定策略对日志进行定期备份等等。

# 二、Log4cPlus安装教程

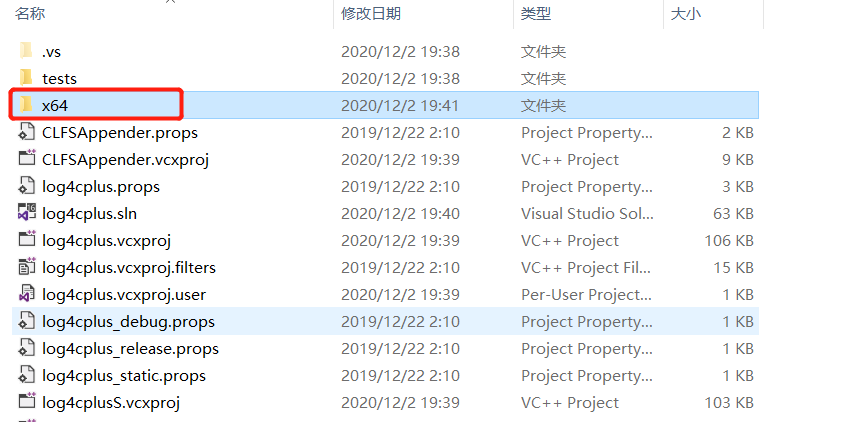
1.到官网https://sourceforge.net/projects/log4cplus/files/log4cplus-stable/，下载最新版log4cplus，将其加压，打开加压后的文件夹，找到log4cplus.sln，将其用Visual Studio打开，log4cplus.sln所在的文件夹位置见图2.1。



**图2.1 log4cplus.sln所在位置**

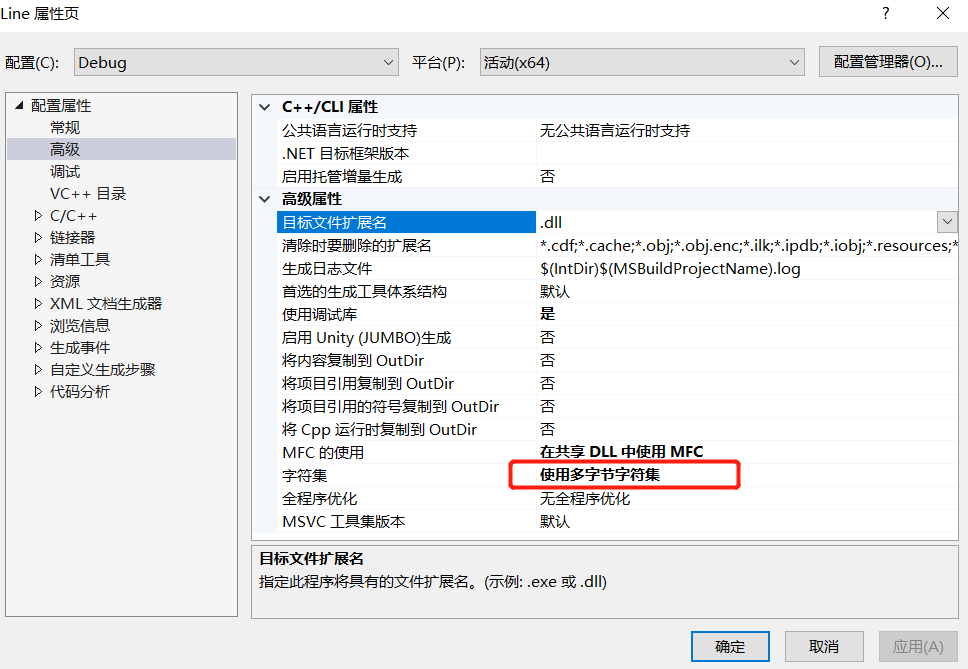
2.打开log4cplus.sln后，找到External Appenders文件夹，将文件夹中的Qt4DebugAppender以及Qt5DebugAppender删除，因为这两个文件需要Qt环境，如果不删除，后面生成解决方案时会出现错误。

3.在Debug x64配置器下生成解决方案，这时在log4cplus.sln同级目录下，会得到一个x64文件夹，如图2.2所示。



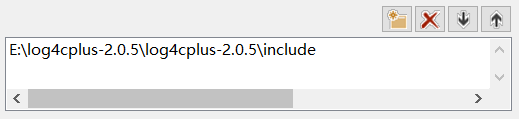
**图2.2 生成的x64文件夹**

4.关闭Visual Studio，新建一个项目，由于新建项目默认是Unicode字符集，所以需要将其修改成多字节字符集，当然也可以修改log4cplus.sln的字符集，只需将项目的字符集与log4cplus.sln的字符集保持一致即可。右键项目，打开属性，选择高级，即可修改字符集，如图2.3所示。



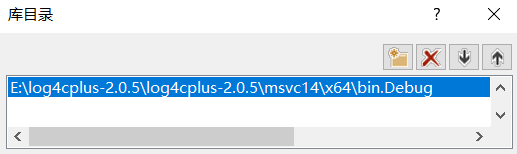
**图2.3 修改项目字符集**

5.配置项目，使其链接到log4cplus库，这里只给出了静态方式链接。点击视图，选择属性管理器，右键Debug|x64，选择属性，然后选择VC++目录，添加包含目录，只需将加压后的include文件包含进来即可，如图2.4所示。



**图2.4 包含目录**

6.添加库目录，添加bin.debug文件夹，如图2.5所示。



**图2.5 添加库目录**

7.然后打开链接器，选择输入，附加依赖项log4cplusSD.lib，然后保存即可。

8.下方给出了一组测试代码，运行结果：在控制台输出INFO-HelloWorld，在程序的文件夹中生成一个log.txt文件。

#include <log4cplus/log4cplus.h>

int main()

{

//用Initializer类进行初始化

log4cplus::Initializer initializer;

//第1步：创建ConsoleAppender和FileAppender(参数app表示内容追加到文件)

log4cplus::SharedAppenderPtr consoleAppender(new log4cplus::ConsoleAppender);

log4cplus::SharedAppenderPtr fileAppender(new log4cplus::FileAppender(

LOG4CPLUS\_TEXT("log.txt"),

std::ios\_base::app

)

);

//第2步：设置Appender的名称和输出格式

//ConsoleAppender使用SimpleLayout

//FileAppender使用PatternLayout

consoleAppender->setName(LOG4CPLUS\_TEXT("console"));

consoleAppender->setLayout(std::unique\_ptr<log4cplus::Layout>(new log4cplus::SimpleLayout()));

fileAppender->setName(LOG4CPLUS\_TEXT("file"));

log4cplus::tstring pattern = LOG4CPLUS\_TEXT("%D{%m/%d/%y %H:%M:%S,%Q} [%t] %-5p %c - %m [%l]%n");

fileAppender->setLayout(std::unique\_ptr<log4cplus::Layout>(new log4cplus::PatternLayout(pattern)));

//第3步：获得一个Logger实例，并设置其日志输出等级阈值

log4cplus::Logger logger = log4cplus::Logger::getInstance(LOG4CPLUS\_TEXT("test"));

logger.setLogLevel(log4cplus::INFO\_LOG\_LEVEL);

//第4步：为Logger实例添加ConsoleAppender和FileAppender

logger.addAppender(consoleAppender);

logger.addAppender(fileAppender);

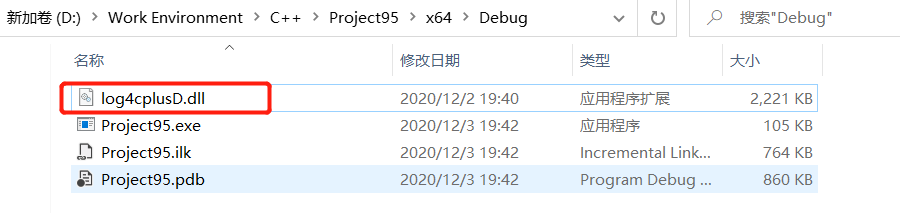
//第5步：使用宏将日志输出

LOG4CPLUS\_INFO(logger, LOG4CPLUS\_TEXT("Hello world"));

return 0;

}

9.运行程序时如果出现找不到log4cplusD.dll的情况，需要将前面生成的x64文件中的log4cplusD.dll复制到当前执行程序的同级目录中，如图2.6所示。



**图2.6 添加log4cplusD.dll**