log4cplus是一款优秀的基于C/C++的开源日志库。log4cplus具有线程安全，不用但心在多线程状态下写日志问题；使用灵活，可通过配置文件设置日志级别下输出位置，还可以在程序运行时动态设置日志输出级别，随心所欲掌控日志的输出；以及多粒度控制的特点。通过将信息划分优先级使其可以面向程序调试、运行、测试、和维护等全生命周期； 可以选择将信息输出到屏幕、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。可以满足大部分开发者对日志系统需求，功能全面

1. 使用log4cplus时，要加上-llog4cplus -lrt -lpthread -lrt(或者 -llog4cplus -lpthread )，才能顺利的编译通过和使用。
2. log4plus支持流式编程。
3. 跨文件使用同一个log4cplus::Logger对象解决办法：

main函数中声明、定义log4cplus::Logger类型对象，使用该对象的文件中使用extern修改该对象即可。

实例程序如下：

main.cxx

#include <log4cplus/logger.h>

#include <log4cplus/configurator.h>

using namespace std;

using namespace log4cplus;

Logger logger = Logger::getRoot();

void TestLog();

int main()

{

cout << "Entering main()..." << endl;

PropertyConfigurator::doConfigure(LOG4CPLUS\_TEXT("log4cplus.properties"));

LOG4CPLUS\_DEBUG(logger, "printMsgs()");

TestLog();

cout << "Exiting main()..." << endl;

return 0;

}

fun.cxx

#include <log4cplus/logger.h>

//外部声明logger，logger定义在main.cxx中

extern log4cplus::Logger logger;

void TestLog()

{

LOG4CPLUS\_ERROR(logger,"TestLog execute ......");

}