glog编译及使用

# 环境

Windows 7 x64 sp1

Visual Studio 2008 SP1

glog-0.3.4.zip

# 编译

解压glog-0.3.4.zip，根目录$(glog)

Visual studio 2008打开$(glog)\google-glog.sln，直接编译即可

libglog为glog动态库

libglog\_static为glog静态lib库

为方便，设置库输出文件为

$(OutDir)\$(ProjectName)\_$(ConfigurationName)\_$(PlatformName).dll

或

$(OutDir)\$(ProjectName)\_$(ConfigurationName)\_$(PlatformName).lib

# 静态链接glog

## 头文件及库文件

准备文件夹$(gloglibs)

将$(glog)\src\windows下的glog文件夹复制到$(gloglibs)

将$(ProjectName)\_$(ConfigurationName)\_$(PlatformName).lib复制到$(gloglibs)

## 项目设置

VS2008新建win32空项目

设置 附加包含目录 和 附加库目录 为$(gloglibs)

链接lib库代码

#ifdef \_DEBUG

#pragma comment(lib, "libglog\_static\_Debug\_Win32")

#else

#pragma comment(lib, "libglog\_static\_Release\_Win32.lib")

#endif // \_DEBUG

包含头文件：

#include <glog/logging.h>

## 使用glog

#include <windows.h>

// avoid ERROR defination in windows.h & logging.h

#undef ERROR

// static link to glog lib

#define GOOGLE\_GLOG\_DLL\_DECL

#include <glog/logging.h>

#include <iostream>

using namespace std;

// atuo link lib

#ifdef \_DEBUG

#pragma comment(lib, "libglog\_static\_Debug\_Win32")

#else

#pragma comment(lib, "libglog\_static\_Release\_Win32.lib")

#endif // \_DEBUG

bool g\_bRunning = true;

unsigned LogTest(int id)

{

int ii = 0;

while (g\_bRunning)

{

stringstream ss;

ss << "log by " << id << " , cnt " << ++ii;

VLOG(10) << " 中文vlog " << ss.str();

LOG(INFO) << " 中文info " << ss.str();

LOG(WARNING) << " 中文warning" << ss.str();

LOG(ERROR) << " 中文error " << ss.str();

LOG\_EVERY\_N(ERROR, 10) << " 中文every10 " << google::COUNTER << "th " << ss.str();

Sleep(1);

}

return 0;

}

int main(int argc, char\*\* argv)

{

// search GLOG\_DEFINE\_ for more FLAGS

// set log path to program path

string str;

str = argv[0];

str = str.substr(0, str.rfind('\\'));

FLAGS\_log\_dir = str;

// max log size MB

FLAGS\_max\_log\_size = 10;

FLAGS\_colorlogtostderr = false;

// all log to stderr (console)

FLAGS\_alsologtostderr = true;

// set log main name

google::InitGoogleLogging(argv[0]);

LOG(INFO) << "this is a info log by main";

int nThread = 1000;

HANDLE\* phThread = new HANDLE[nThread];

for (int ii = 0; ii < nThread; ++ii)

{

phThread[ii] = CreateThread(0, 0, (LPTHREAD\_START\_ROUTINE)LogTest, (LPVOID)ii, 0, 0);

}

char ch = getchar();

g\_bRunning = false;

for (int ii = 0; ii < nThread; ++ii)

{

WaitForSingleObject(phThread[ii], INFINITE);

}

// clear glog global memory

google::ShutdownGoogleLogging();

return 0;

}