# **[visual studio运行时库MT、MTd、MD、MDd的研究](https://www.cnblogs.com/ShaneZhang/p/3480502.html)**

在开发window程序是经常会遇到编译好好的程序拿到另一台机器上面无法运行的情况，这一般是由于另一台机器上面没有安装响应的运行时库导致的，那么这个与编译选项MT、MTd、MD、MDd有什么关系呢？这是msdn上面的解释：

**MT**：mutithread，多线程库，编译器会从运行时库里面选择多线程静态连接库来解释程序中的代码，即连接LIBCMT.lib库

**MTd**：mutithread+debug，多线程调试版，连接LIBMITD.lib库

**MD**：MT+DLL，多线程动态库，连接MSVCRT.lib库，这是个导入库，对应动态库为MSVCRT.dll

**MDd**： MT+DLL+debug，多线程动态调试库，连接MSVCRTD.lib库，对应动态库为MSVCRTD.dll

开发多线程程序时（单线程本文不做讨论），需要选择MT、MTd、MD、MDd其中的一个。

对于MT/MTd，由于连接运行时库是LIBCMT.lib/LIBCMTD.lib，这两个库是静态库，所以此种方式编译的程序，移到另一台机器上面也可以正常运行。

但是对于MD/MDd，连接的是动态库，所以如果另一台机器上没有MSVCRT.dll/MSVCRTD.dll时，就提示缺少动态库这样的错误。

曾经犯这样的错误，以为以MT/MTd方式编译，程序对所有的库都是静态链接的，其实错了，它只能决定运行时库是动态链接还是静态链接，对用户自己写的库或其他第三方库，其连接方式取决于代码（显示连接动态库Loadlibrary）或所提供的lib文件（为导入库还是静态库），移动程序到别的机器上时，还是要带上所需要的动态库的。

来看一个例子，编译一个静态库和一个动态库，均实现两个整数相加的功能：

// adds.h// add后面加个s代表静态库#pragma onceint add(int,int);

[IMG_256](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

// adds.cpp// 静态库

#include "adds.h"int add(int a, int b)

{

return a+b;

}

以上，运行时库选择MTd，编译成静态库adds.lib

[IMG_257](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

[IMG_258](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

// addd.h// add后面加d代表动态库#pragma once

#ifndef MYLIB\_API#define MYLIB\_API \_declspec(dllexport)#endif

MYLIB\_API int add(int,int);

[IMG_259](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

[IMG_260](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

// addd.cpp// 动态库

#include "addd.h"int add(int a, int b)

{

return a+b;

}以上，运行时库选择MTd，编译成动态库addd.lib, addd.dll

[IMG_261](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

[IMG_262](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

// test.cpp// 测试程序

#include <iostream>

// 测试静态库，此处为1，测试动态库，此处为0#define TEST\_STATIC\_LINK 1

#if TEST\_STATIC\_LINK

#include <adds.h>#else

#define MYLIB\_API \_declspec(dllimport)

#include "addd.h"#endif

using namespace std;

int main()

{

cout << add(2,3) << endl;

return 0;

}测试程序以MTd编译

[IMG_263](https://www.cnblogs.com/ShaneZhang/p/javascript:void(0);)

1. 测试静态库，TEST\_STATIC\_LINK 定义为1，提供adds.lib，生成可执行文件，移动到另一台机器上可以运行，因为测试程序和adds.lib均静态连接运行时库

2. 测试动态库，TEST\_STATIC\_LINK 定义为0，提供addd.lib，生成可执行文件，移动到另一台机器上可以运行，但需要addd.dll,因为addd库静态连接运行时库,测试程序静态连接运行时库，动态连接addd库

在上面的例子中add库和测试程序均选择MTd运行时库，若不一致会导致一些编译连接错误，让新手不着头脑。

比如adds选择MDd，连接将会出现这样的错误：

1>正在链接...

1>MSVCRTD.lib(ti\_inst.obj) : error LNK2005: "private: \_\_thiscall type\_info::type\_info(class type\_info const &)" ([??0type\_info@@AAE@ABV0@@Z](mailto:??0type_info@@AAE@ABV0@@Z)) 已经在LIBCMTD.lib(typinfo.obj) 中定义

1>MSVCRTD.lib(ti\_inst.obj) : error LNK2005: "private: class type\_info & \_\_thiscall type\_info::operator=(class type\_info const &)" ([??4type\_info@@AAEAAV0@ABV0@@Z](mailto:??4type_info@@AAEAAV0@ABV0@@Z)) 已经在LIBCMTD.lib(typinfo.obj) 中定义

**即一个程序中混合了不同的运行时库（静态库和动态库，调试库和非调试库），可能会产生冲突，所以一个程序中应该使用相同的运行时库。**

原帖地址<http://blog.csdn.net/ybxuwei/article/details/9095067>

<https://blog.csdn.net/cjqpker/article/details/46713791>