

VS 编译的 MT 和 MD 区别

Dezeming Family

2023 年 4 月 16 日

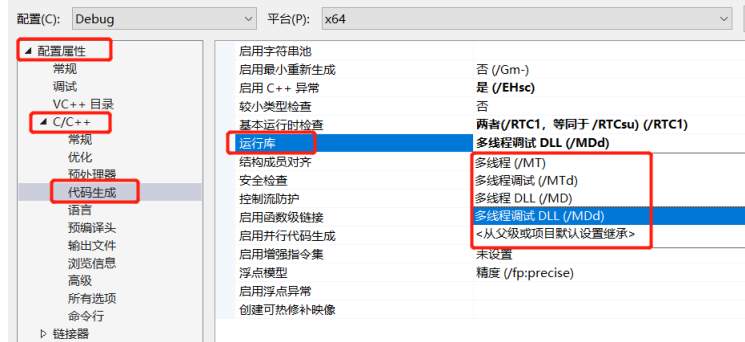
DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [<https://dezeming.top/>] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

目录

一 基本介绍	1
二 基本原理	1
2.1 动态链接库和静态链接库	1
2.2 C/C++ 运行时库	1
2.3 VS 中编译时的选项	2
2.4 I/O 流和 STL 库	2
三 小结	3
参考文献	3

一 基本介绍

在 Visual Studio 中，右键项目名称，打开项目属性页，选择“配置属性”->“C/C++”->“代码生成”->“运行库”即可对运行库类型进行设置：



这里运行库类型主要包含四种：多线程 (/MT)、多线程调试 (/MTd)、多线程 DLL(/MD)、多线程调试 (/MDd)。其中带有 d 的类型是项目的调试版本，不带有 d 的类型是项目的 Release 版本。这些类型之间有什么区别和联系呢，这就是本文要介绍的。

二 基本原理

网上很多文章都没有说清楚动态库和静态库的根本区别，也没有说明动态库同样需要 lib 这件事，因此本文会先花一两个小节来详细描述相关知识点。

2.1 动态链接库和静态链接库

为了方便有不同需求的客户使用，C++ 编译成库时，可以编译为动态链接库 dll 版本和静态链接库 lib 版本。

lib 有两种，第一种是包含了函数所在的 dll 文件和文件中函数位置的信息（入口信息），库的代码由运行时加载在进程空间中的 dll 提供，称为动态链接库 dynamic link library。

第二种 lib 可以包含函数代码本身，在编译时直接将代码加入程序当中，称为静态链接库 static link library。

共有两种链接方式：（1）动态链接使用动态链接库，允许可执行模块（一般是.exe 文件，也可以是引用其他库的.dll 文件）仅包含在运行时定位 dll 函数的可执行代码所需的信息。（2）静态链接使用静态链接库，链接器从静态链接库 lib 获取所有被引用函数，并将库同代码一起放到可执行文件中。

所以需要注意的是，动态链接也得需要 lib 文件，用于提供所需的 dll 引用和链接、函数入口等信息。

2.2 C/C++ 运行时库

为了提高 C 语言的开发效率，C 标准定义了一系列常用的函数，称为 C 库函数。C 标准仅仅定义了函数原型，并没有提供实现。因此这个任务留给了各个支持 C 语言标准的编译器。

每个编译器通常实现了标准 C 的超集，称为 C 运行时库（C Run Time Library），简称 CRT。对于 VC++ 编译器来说，它提供的 CRT 库支持 C 标准定义的标准 C 函数，同时也有一些专门针对 Windows 系统特别设计的函数。

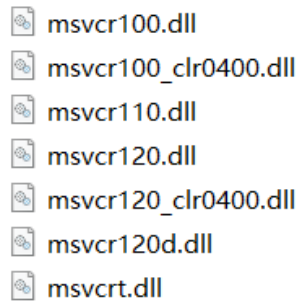
与 C 语言类似，C++ 也定义了自己的标准，同时提供相关支持库，称为 C++ 运行时库或 C++ 标准库。由于 C++ 对 C 的兼容性，C++ 标准库包括了 C 标准库。除了 C++ 运行时库之外，还有包含了 I/O 流和标准模板库 STL 的库。

VC++ 完美的支持 C 和 C++ 标准，因此也就按照 C 和 C++ 标准定义的函数原型实现了上述运行时库。

VC++ 分别实现了动态链接库 dll 版本和静态链接库 lib 版本。同时为了支持程序调试且不影响程序的性能，又分别提供了对应的调试版本。调试版本的名称在 Release 版本名称后添加了字母 d。Windows

环境下，libcmtd.lib 是 VC 提供的多线程静态运行时库，libcmtd.lib 是其 Debug 版本；msvcrt.lib 是多线程动态运行时库，msvcrt.lib 是其对应的 Debug 版本，此外，动态库对应的 dll 版本名称根据编译器不同而不同。

比如，VC6.0 使用的 CRT 库的 dll 版本在 msvcrt.dll（也是系统自带的运行时库 dll 版）中实现；我们打开目录“C:/Windows/System32”，能看到该目录下一般包含有很多种 msvcrt 库：



VS2008 使用的 CRT 库的 dll 版本在 msvcr90.dll 中实现，对应调试版本为 msvcr90d.dll。VS2010 使用的 CRT 库的 DLL 版本在 msvcr100.dll 中实现，对应调试版本为 msvcr100d.dll。msvcr120.dll 是 VS2013 的运行库，对应调试版本为 msvcr120d.dll。

动态 dll 和静态 lib 的 C 和 C++ 运行库的优缺点：因为静态版必须把 C 和 C++ 运行库复制到目标程序中 (.exe)，所以产生的可执行文件会比较大。同时对于使用多个模块的大型软件来说，如果每个模块均选择静态链接 C 或 C++ 运行库，在程序运行时就会存在多个运行库。在链接时也会出现重复定义的问题。

使用 dll 版的 C 和 C++ 运行库，程序在运行时动态的加载对应的 dll。程序本身体积变小。但一个很大的问题是，一旦找不到对应 dll，程序将无法运行（比如所要移植的电脑没有安装 VC++，有些时候运行程序经常会遇到“找不到 msvcrt.dll，重新安装程序以解决”这种报错）。

2.3 VS 中编译时的选项

/MT、/MTd、/MD、/MDd 四个编译选项，分别代表多线程、多线程调试、多线程 DLL、多线程调试 DLL。编译时引用的 lib 分别为 libcmtd.lib、libcmtd.lib、msvcrt.lib 和 msvcrt.lib。

/MT：是 multithread-static version，是多线程静态版本的意思，项目会使用运行时库的多线程静态版本，编译器会将 libcmtd.lib 放入.obj 文件中，以便链接器使用 libcmtd.lib 解析外部符号。

/MTd：定义了 _DEBUG 和 /MT，是 /MT 类型的 debug 版本。

/MD：是 multithread-dll version，项目会使用运行时库的多线程动态 dll 版本，编译器会将 msvcrt.lib 放入.obj 文件中，在编译项目时，它会静态链接 msvcrt.lib，但在实际运行过程中，它会链接使用对应的 dll（比如 msvcr120.dll）。

MDd：是定义了 _DEBUG 和 /MD，是 /MD 类型的 debug 版本。

因此，我们在编译时，对应的库一定要配套一致。比如你用 CMake 生成工程文件时，若 CMake 是用 /MT 类型生成的，则它所调用的运行时库为 libcmtd.lib；后面生成工程时，你必须也选择 /MT 类型，否则如果选择 /MD 类型时，工程在链接时会调用 msvcrt.lib 和 msvcrt.dll，显然两次调用不同的运行时库，会出现重定义的错误：在编译包含多个模块的程序时，经常会遇到 libcmtd.lib 和 msvcrt.lib 冲突，这就是因为当使用外部库，引用的库用的编译选项和当前项目用到的编译选项不同，分别用了相类似功能的不同 lib，符号当然也有很多重复。

总的来说，/MD 类型表示运行时库不集成，生成的文件小，程序运行时动态加载对应的 dll；/MT 类型表示运行时库集成，生成的文件较大，在链接时将 C/C++ 运行时库集成到程序中。

2.4 I/O 流和 STL 库

对于 C++ 标准库中的 I/O 流和 STL，静态链接版实现在 libcpmt.lib 中，对应的调试版本为 libcpmtd.lib。

动态链接库的名称为 msvcprt.lib，对应 msvcprt.dll、msvcprt100.dll、msvcprt110.dll、msvcprt120.dll、msvcprt140.dll 等多个动态链接库版本。

三 小结

到这里，关于不同的链接选项方面的介绍就暂时到此为止了。

Visual Studio 中还有很多与编译配置有关的内容，只有把这些内容都搞清楚，才不会在编译时需要各种古怪的、令人束手无策的 Bug。

参考文献

- [1] <https://zhuanlan.zhihu.com/p/68846776>
- [2] <https://zhuanlan.zhihu.com/p/375482880>
- [3] <https://www.cnblogs.com/cswuyg/archive/2012/02/03/2336424.html>
- [4] <https://blog.csdn.net/wowocpp/article/details/122130532>
- [5] <https://www.shuzhiduo.com/topic/libcmt-lib>