

debugrelease区别:Debug和Release之本质区别

疯狂代码 <http://www.CrazyCoder.cn/> [j:http://www.CrazyCoder.cn/VC/Article12664.html](http://www.CrazyCoder.cn/VC/Article12664.html)

本文主要包含如下内容：

1. Debug 和 Release 编译方式的本质区别
2. 哪些情况下 Release 版会出错
3. 怎样“调试” Release 版的程序

关于Debug和Release之本质区别的讨论 一、Debug 和 Release 编译方式的本质区别 Debug 通常称为调试版本，它包含调试信息，并且不作任何优化，便于程序员调试程序。Release 称为发布版本，它往往是进行了各种优化，使得程序在代码大小和运行速度上都是最优的，以使用户很好地使用。

Debug 和 Release 的真正秘密，在于一组编译选项。下面列出了分别针对二者的选项（当然除此之外还有其他一些，如/Fd /Fo，但区别并不重要，通常他们也不会引起 Release 版错误，在此不讨论） Debug 版本：

/MDd /MLd 或 /MTd 使用 Debug runtime library(调试版本的运行时刻函数库)

/Od 关闭优化开关

/D "_DEBUG" 相当于 #define _DEBUG,打开编译调试代码开关(主要针对 assert函数)

/ZI 创建 Edit and continue(编辑继续)数据库，这样在调试过程中如果修改了源代码不需重新编译

/GZ 可以帮助捕获内存错误

/Gm 打开最小化重链接开关，减少链接时间 Release 版本：

/MD /ML 或 /MT 使用发布版本的运行时刻函数库

/O1 或 /O2 优化开关，使程序最小或最快

/D "NDEBUG" 关闭条件编译调试代码开关(即不编译assert函数)

/GF 合并重复的字符串，并将字符串常量放到只读内存，防止

被修改 实际上，Debug 和 Release 并没有本质的界限，他们只是一组编译选项的集合，编译器只是按照预定的选项行动。事实上，我们甚至可以修改这些选项，从而得到优化过的调试版本或是带跟踪语句的发布版本。

二、哪些情况下 Release 版会出错 有了上面的介绍，我们再来逐个对照这些选项看看 Release 版错误是怎样产生的 1. Runtime Library：链接哪种运行时刻函数库通常只对程序的性能产生影响。调试版本的 Runtime Library 包含了调试信息，并采用了一些保护机制以帮助发现错误，因此性能不如发布版本。编译器提供的 Runtime Library 通常很稳定，不会造成 Release 版错误；倒是由于 Debug 的 Runtime Library 加强了对错误的检测，如堆内存分配，有时会出现 Debug 有错但 Release 正常的现象。应当指出的是，如果 Debug 有错，即使 Release 正常，程序肯定是有 Bug 的，只不过可能是 Release 版的某次运行没有表现出来而已。

2008-9-24 12:11:04

疯狂代码 <http://www.CrazyCoder.cn/>