

VC下Debug和Release区别 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/471/2021\\_2022\\_VC\\_E4\\_B8\\_8BDebug\\_c67\\_471203.htm](https://www.100test.com/kao_ti2020/471/2021_2022_VC_E4_B8_8BDebug_c67_471203.htm) 最近写代码过程中，发现 Debug 下运行正常，Release 下就会出现各种问题，百思不得其解，而Release 下又无法进行调试，于是只能采用printf方式逐步定位到问题所在处，才发现原来是给定的一个数组未初始化，导致后面处理异常。网上查找了些资料，在这罗列汇总下，做为备忘

~ 一、Debug 和 Release 的区别 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 的真正区别，在于一组编译选项。

Debug 版本 参数 含义 /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

Release 版本 参数含义 /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

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。 相关经验：转

自<http://dev.csdn.net/article/17/17068.shtm> 1. 变量。大家都知道

， debug跟release在初始化变量时所做的操作是不同的， debug是将每个字节位都赋成0xcc(注1)， 而release的赋值近似于随机(我想是直接从内存中分配的， 没有初始化过)。这样就明确了， 如果你的程序中的某个变量没被初始化就被引用， 就很有可能出现异常：用作控制变量将导致流程导向不一致；用作数组下标将会使程序崩溃；更加可能是造成其他变量的不准确而引起其他的错误。所以在声明变量后马上对其初始化一个默认的值是最简单有效的办法， 否则项目大了你找都没地方找。代码存在错误在debug方式下可能会忽略而不被察觉到， 如debug方式下数组越界也大多不会出错， 在release中就暴露出来了， 这个找起来就比较难了:( 还是自己多加注意吧 呵呵， 就是我犯的问题 ~ ~

2. 自定义消息的消息参数。  
MFC为我们提供了很好的消息机制， 更增加了自定义消息， 好处我就不必多说了。这也存在debug跟release的问题吗？答案是肯定的。在自定义消息的函数体声明时， 时常会看到这样的写法：`afx_msg LRESULT OnMessageOwn()`。 Debug情况下一般不会有任问题， 而当你在Release下且多线程或进程间使用了消息传递时就会导致无效句柄之类的错误。导致这个错误直接原因是消息体的参数没有添加， 即应该写成

：`afx_msg LRESULT OnMessageOwn(WPARAM wparam, LPARAM lparam)`。(注2) 3. release模式下不出错， 但debug模式下报错。 这种情况下大多也是因为代码书写不正确引起的， 查看MFC的源码， 可以发现好多ASSERT的语句(断言)， 这个宏只是在debug模式下才有效， 那么就清楚了， release版不报错是忽略了错误而不是没有错误， 这可能存在很大的隐患， 因为是Debug模式下， 比较方便调试， 好好的检查自己的代码

，再此就不多说了。4. ASSERT, VERIFY, TRACE.....调试宏这种情况很容易解释。举个例子：请在VC下输入ASSERT然后选中按F12跳到宏定义的地方，这里你就能够发现Debug中ASSERT要执行AfxAssertFailedLine，而Release下的宏定义却为"#define ASSERT(f) ((void)0)"。所以注意在这些调试宏的语句不要用程序相关变量如i写操作的语句。VERIFY是个例外，"#define VERIFY(f) ((void)(f))"，即执行，这里的作用就不多追究了，有兴趣可自己研究:)。总结：Debug与Release不同的问题在刚开始编写代码时会经常发生，99%是因为你的代码书写错误而导致的，所以不要动不动就说系统问题或编译器问题，努力找找自己的原因才是根本。我从前就常常遇到这情况，经历过一次次的教训后我就开始注意了，现在我所写过的代码我已经好久没遇到这种问题了。下面是几个避免的方面，即使没有这种问题也应注意一下：1. 注意变量的初始化，尤其是指针变量，数组变量的初始化(很大的情况下另作考虑了)。2. 自定义消息及其他声明的标准写法3. 使用调试宏时使用后最好注释掉4. 尽量使用try - catch(...)5. 尽量使用模块，不但表达清楚而且方便调试。注1：debug版初始化成0xcc是因为0xcc在x86下是一条int 3单步中断指令，这样程序如果跑飞了遇到0xcc就会停下来，这和单片机编程时一般将没用的代码空间填入jmp 0000语句是一样地

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)