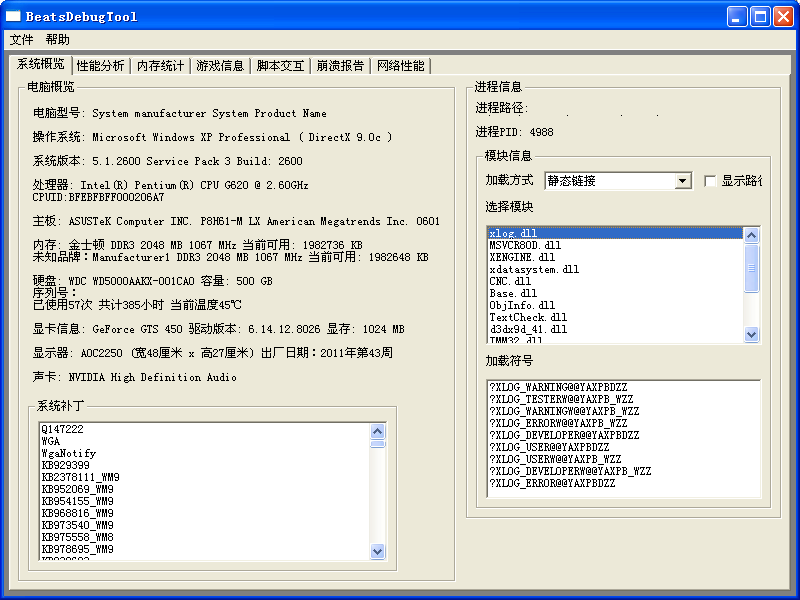


选择要内存监控的模块（支持多选），然后点启动

如果只选cgame.exe，内存监控效率会好很多，支持自动刷新

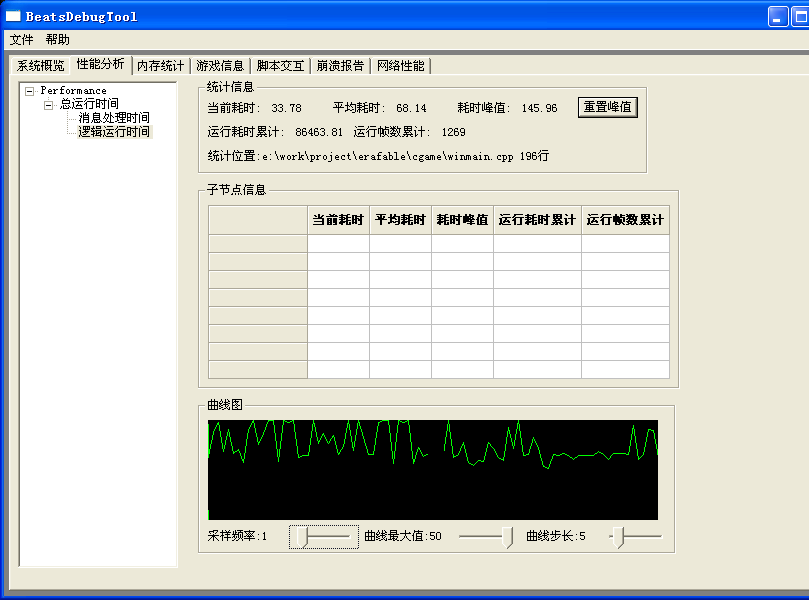
如果另外还选了xengine,因为数量级会到20万，所以很卡，但程序会自动转为手动刷新

一个程序自动刷新次数越频繁，则其越容易造成内存碎片，我们的程序对堆的操作应该尽量稳定。



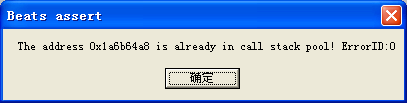
系统信息

可以通过选中静态链接模块查看其中符号，得知每个模块引用的原因，从而剔除一些不必要的动态库。

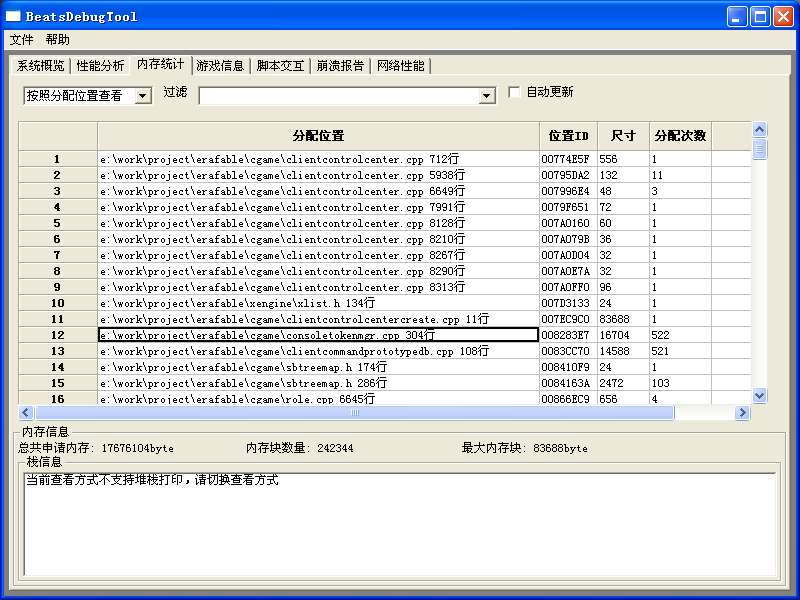


性能分析，通过选择左边的节点，可以知道每个分支的信息和消耗曲线。

具体的使用方法可以参考winmain.cpp

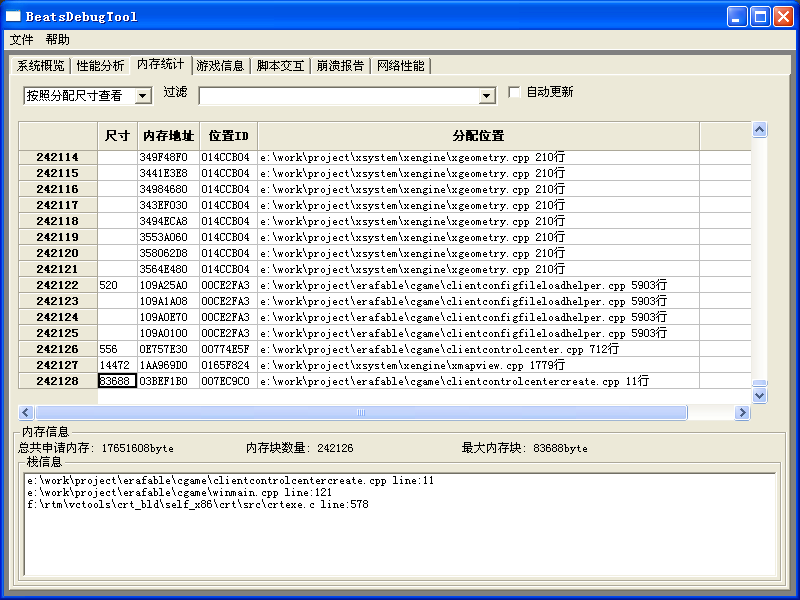


出现这样的对话框一律忽略，因为目前还没有支持C语言的内存操作符，或者说目标模块出现了 堆损坏。



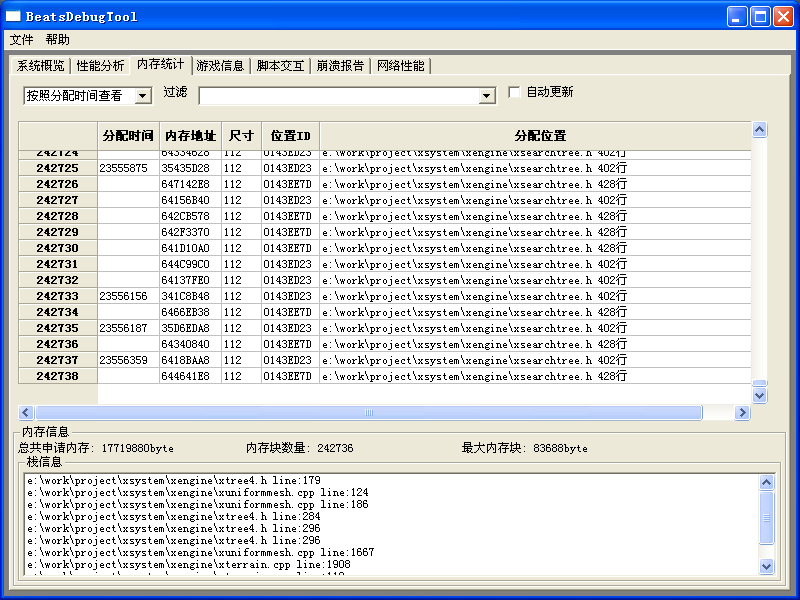
按照位置查看

如第12记录，在该位置执行了522次，一共分配内存16704字节。因为这里是按照位置分布进行统计，所以不支持打印选择条目的堆栈，另外因为UI刷新需要的时间最少，这个也是最不卡的查看方式。



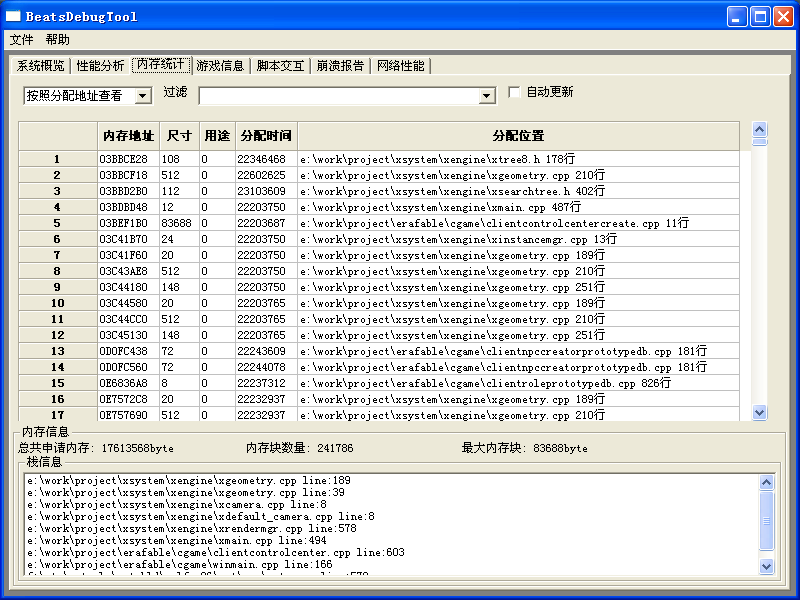
按照分配尺寸查看

如图所示，因为这里要打印24万条记录，所以一旦选中就会卡个几十秒。拉到最下面可以看到，我们在一个地方申请了83k内存，这也是我们客户端申请尺寸最大的一个内存块。



按照分配时间查看

这里可以查看到我们最近一次进行内存操作的位置，如果这个列表不断的在刷新，表明内存管理还需要改进，如果内存块数量或者总尺寸在不断增加，说明当前运行的流程包含了隐含的性能问题，比较适合测试QC报告的根据时间流逝导致性能下降的bug。



按照分配地址查看

这里是在枚举所有的内存块地址。如果以后有对损坏，或者任何非法内存操作，可以到这里比对那块内存是在什么位置分配的，从而迅速的找准位置。

安装方法：

1、 将dll拷贝到运行目录

2、将winmain.cpp与客户端Cgame工程中的winmian.cpp替换即可。

源代码可以在我的共享文件夹里看到。