Exception in thread "main" java.lang.OutOfMemoryError: Java heap space解决方法

Exception in thread "main" java.lang.OutOfMemoryError: Java heap space解决方法

2011年09月16日 星期五 下午 3:10

|  |
| --- |
| **问题描述** Exception in thread "main" **java.lang.OutOfMemoryError: Java heap space**  **解决方案[转]** 一直都知道可以设置jvm heap大小，一直用eclipse写/调试java程序。一直用命令行or console加参数跑程序。现象：在eclipse的配置文件eclipse.ini中设置**-vmargs -Xms500m -Xmx1024m**，在eclipse中直接run 或者debug某些耗内存的程序时依然出现java.lang.OutOfMemoryError: Java Heap Space错误，即通常认为的内存不足，java虚拟机内存不够用。而在命令行加这些参数则有效果，不会出错。这说明一个问题，这些参数根本没有起作用。今天需要在eclipse里调试程序，还没到需要调试的地方就heap error了，在网上搜了很多地方，得到了最终的答案： 选中被运行的类，点击菜单‘**run->run...**’，选择(x)=Argument标签页下的**vm arguments**框里 输入 **-Xmx800m**, 保存运行。 原来还**需要对每个project单独设置**，汗...   有三种可能导致OutOfMemoryError。首先是，此JVM有真实的内存泄漏，导致此JVM堆在内部实现时产生了一个Bug。这极不可靠。所有JVM都经过充分的测试，并且，如果有人发现这种bug，它将绝对是最高的优先级。因此你可以非常宽心地排除这种可能性。  第二种可能的OutOfMemoryError原因只不过是，你没有为你的应用程序运行时给予足够多的可用内存。这种情况，有两种可能的方案，或者增加 JVM堆可用大小，或者减少你的应用程序所需的内存总量。提高JVM可用堆大小可以简单的使用JVM的 -Xmx 参数。假如你将此参数设置尽可能的大（可用内存极限不要超过系统物理内存，否则你的应用程序将分页并暂停），仍然有以上所提到的内存问题，那么，你需要减 少你的应用程序所可能用到内存总量。减少应用程序内存可能是简单的，你可能允许一些集合过大，例如使用了许多大的缓冲区。或者它过于复杂，要求你重新实现 一些类，乃至重新设计应用程序。  读者 Jams Stauffer 指出有些JVM（例如 sun的 JVMs），还有一个“Perm”参数用来处理JVM结构与类对象。如果你正在使用一个数量非常巨大的类集，它有可能运行在"Perm"空间之外，然后你 需要增加此空间的大小，例如，sun的JVM使用 -XX:PermSize 与 -XX:MaxPermSize 选项。  第三种导致OutOfMemoryError最为常见，无心的对象引用保持。你没有明确无误的释放对象，以致于你的堆增长再增长，直到你没有额外的空间。  处理OutOfMemoryError:  是JVM内部的BUG？不太可能。如果是，这是优先级最高的BUG（为什么还没有人发现它，而你碰到了？）。  没有足够的内存分配给实际运行的应用程序？两种选择：使用-Xmx参数增加堆的最大使用内存（或者使用-XX:MaxPermSize参数增加Perm空 间大小）;　或者使用更小的集合/缓冲区/表空间/对象.....，以减少所需要的内存总量，也就是说，可以调整对象大小，重新设计与重新实现你的应用程 序。  无心的对象引用保持？找到保持这些无意引用的源对象，改变它并释放这些对象。在IBM开发者社区的文章纲要式的揭示了这样一个通用的处理过程。这个过程主 要是等到应用程序到达恒定状态－－你将期望最多的新创建的对象是临时对象，并且可以被垃圾收集器收集。这常常是在应用程序所有的初始化工作完成之后。  强迫垃圾收集，获得一个堆的对象快照。 做任何工作可能正在导到无意的对象引用保持。 强迫另一次垃圾收集并获得第二次堆的对象快照。 比较这两个快照，观察从第一个快照到第二个快照哪些对象在数量上有所增加。因为你在快照之前强迫垃圾收集，剩下的将是所有被应用程序引用的对象，比较两个快照将准确的标识那些新创建的、保留在应用程序里的对象。 根据你对应用程序的认识，决定两个快照比较中，哪些对象正在无意的保持对象引用。 跟踪前导引用，找到哪些对象正在引用这些无意的保持对象，直到你找到导致此问题的源对象  启动虚拟机的时候，加上一个参数：-Xms800m -Xmx800m就好了  **-Xms <size>  设置JVM初始化堆内存大小   -Xmx <size>  设置JVM最大的堆内存大小**  如果是应用程序，则：java -Xms800m -Xmx800m 你的类名  如果是tomcat之类的web服务器，在这个服务器的启动文件后面加上这个参数即可。   另外设置环境变量 JAVA\_OPTS="-server -Xms800m -Xmx800m -XX:PermSize=64M -XX:MaxNewSize=256m -XX:MaxPermSize=128m -Djava.awt.headless=true " |