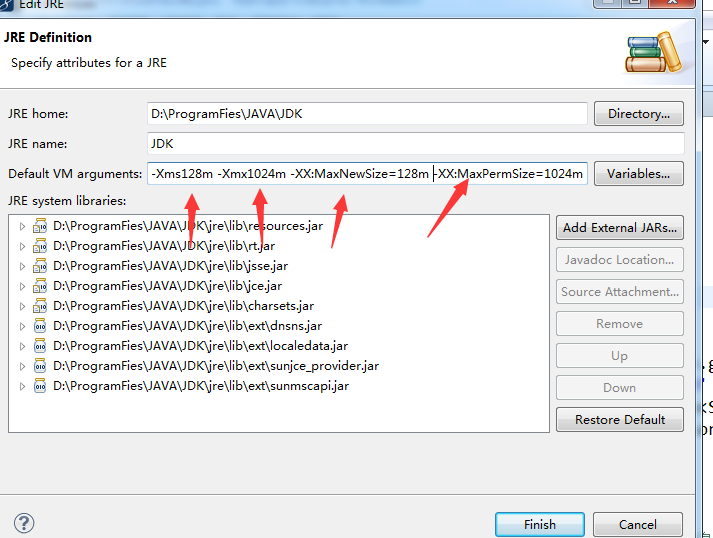
<http://blog.csdn.net/kok1992/article/details/44774967>



# -x，堆区（程序员）

**1、-Xms ：表示java虚拟机堆区内存初始内存分配的大小，通常为操作系统可用内存的1/64大小即可，但仍需按照实际情况进行分配。有可能真的按照这样的一个规则分配时，设计出的软件还没有能够运行得起来就挂了。**

-Xms128m：虚拟机占用系统的最小内存  
    2、-Xmx： 表示java虚拟机堆区内存可被分配的最大上限，通常为操作系统可用内存的1/4大小。但是开发过程中，通常会将 -Xms 与 -Xmx两个参数的配置相同的值，其目的是为了能够在java垃圾回收机制清理完堆区后不需要重新分隔计算堆区的大小而浪费资源。

-Xmx256m：虚拟机占用系统的最大内存

## 一般来讲对于堆区的内存分配只需要对上述两个参数进行合理配置即可，但是如果想要进行更加精细的分配还可以对堆区内存进一步的细化，那就要用到下面的三个参数了-XX:newSize、-XX:MaxnewSize、-Xmn。当然这源于对堆区的进一步细化分：新生代、中生代、老生代。java中每新new一个对象所占用的内存空间就是新生代的空间，当java垃圾回收机制对堆区进行资源回收后，那些新生代中没有被回收的资源将被转移到中生代，中生代的被转移到老生代。而接下来要讲述的三个参数是用来控制新生代内存大小的。

**1、**-XX:newSize：表示新生代初始内存的大小，应该小于 -Xms的值；

**2、**-XX:MaxnewSize：表示新生代可被分配的内存的最大上限；当然这个值应该小于 -Xmx的值；

### -XX:MaxnewSize128m：新生代可被分配的，应该小于 -Xms的值

    **3、**-Xmn：至于这个参数则是对 -XX:newSize、-XX:MaxnewSize两个参数的同时配置，也就是说如果通过-Xmn来配置新生代的内存大小，那么-XX:newSize = -XX:MaxnewSize = -Xmn，虽然会很方便，但需要注意的是这个参数是在JDK1.4版本以后才使用的。

# XX、非堆区 （栈区）

# 上面所述即为java虚拟机对外提供的可配置堆区的参数，接下来讲述java虚拟机对非堆区内存配置的两个参数：

 **1、**-XX:PermSize：表示非堆区初始内存分配大小，其缩写为permanent size（持久化内存）

### -XX:PermSize：最小栈内存大小。一般报内存不足时,都是说这个太小,堆空间剩余小于5%就会警告,建议把这个稍微设大一点,不过要视自己机器内存大小来设置     2、-XX:MaxPermSize：表示对非堆区分配的内存的最大上限。

-XX:MaxPermSize1024m：最大栈内存大小。这个也适当大些