**Android合流白皮书（内存篇）**

**标准：**

1.Android手持设备最大进程内存阀值的50%

2.不允许出现内存泄漏

3.OOM要有异常处理机制

**常用工具：**

1.       DDMS

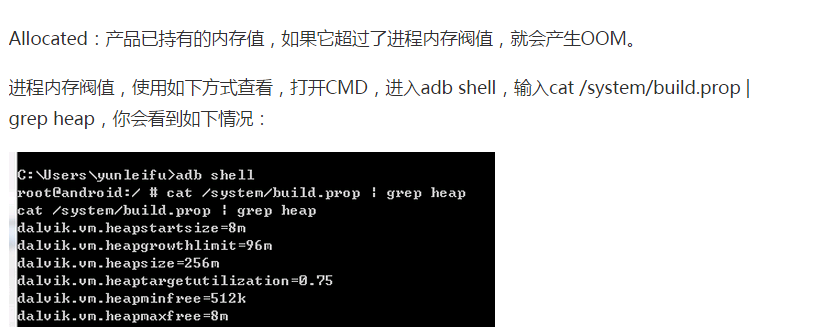
2.       MAT

**执行方法：**

**50%内存阀值监控**

【查看设备最大进程内存阀值的方法】

方法一：



当部分手机不支持grep过滤时，使用：cat /system/build.prop，在结果中查看。

截图当中的heapgrowthlimit即为内存阀值。

方法二:

使用工具GetHeapSize，在手机上直接查看。

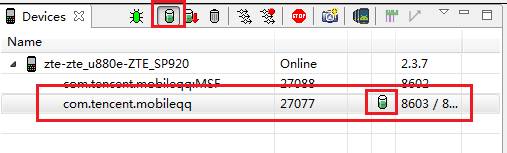
**查看法**

【级别：必须执行】

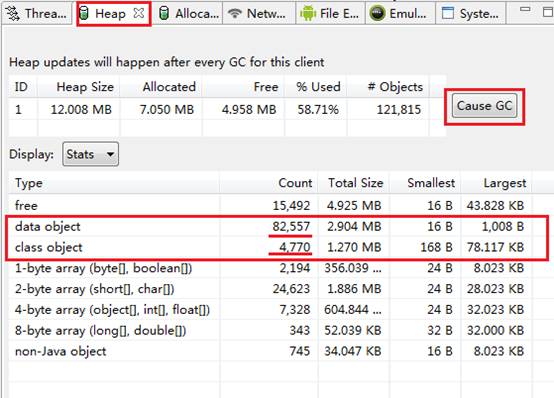
使用这个步骤基本上就能帮助测试，发现和监控50%进程内存阀值红线的，触及情况。

【方法】

使用DDMS的设备进程管理器（只有调试版apk所产生的进程可以显示在这个列表中，开发版的os会显示所有进程，不管是否是调试版apk）



对列表中选中进程，做update heap操作，就可以在heap标签下看到当前进程所占内存



如果列表是空，看不到这些数据，请点击【cause GC】按钮

可以注重看Allocated字段，和data object字段

Allocated：当前占用内存大小（如果此项超过最大进程内存阀值的50%，点击【cause GC】无法降低，就触及合入红线了）

data object：进程中实例个数

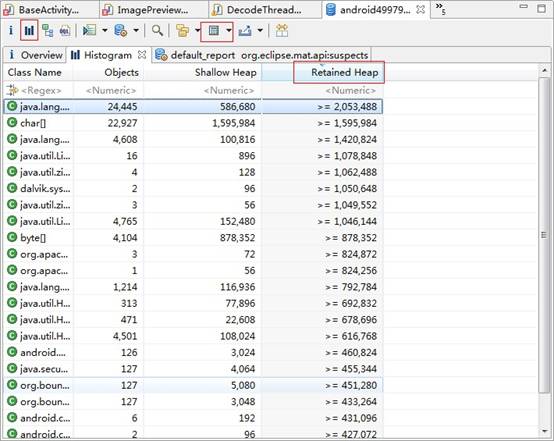
开发分析法

【级别：建议开发执行】

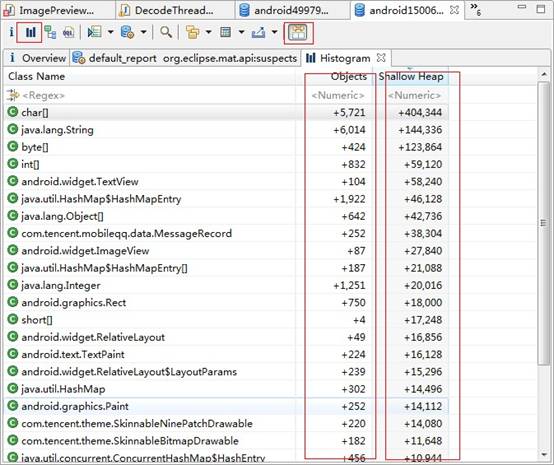
           执行这个步骤可以定量，定性内存问题发生的原因，机制。但使用过程比较繁琐，需要多加训练，建议开启培训机制

【方法】

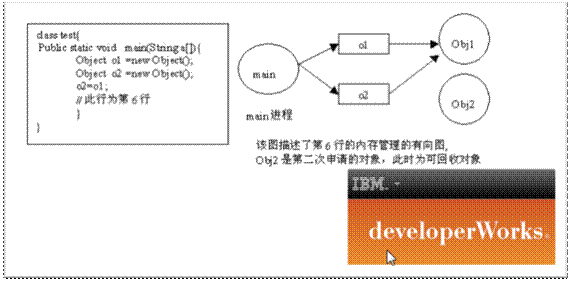
排列法：使用MAT的统计视图排列功能，排列出内存占用最大的对象，然后逐个分析引用关系。



对比法：使用MAT统计视图的对比功能，对比两个hprof文件，发现增量部分，然后通过统治者视图分析引用关系。（可以参看木糖醇的【内存杂谈】）



代码静态分析法：直接命中代码，由开发主导，看代码画出对象向量图，分析引用关系。



**内存泄漏**

使用【查看法】，反复使用相同功能，如果【data object】持续上升，点击【cause GC】也无法回降，就是内存泄漏了，这个是不允许的，触发了合入标准红线

**OOM异常处理机制**

OutOfMemoryError的异常处理机制，应包括

1.       释放cache内存（比如使用list，vector等所做的缓存，执行clear操作）

2.       调用系统GC方法

3.       打印log，包括模块名称，异常出现的方法名称

4.       重新执行异常前操作，如果依然出现内存不足，直接crash，使用异常上报机制，上报异常。

**案例：**

**切换帐号内存泄漏**

使用【查看法】，发现切换帐号过程中【data object】持续上升，点击【cause GC】也无法回降，用MAT抓取dump，分析结果

