# Layout

## 1, adb shell dumpsys window (displays)

<http://quanminchaoren.iteye.com/blog/1250299>

显示所有界面的window的详细信息

WindowManager.java

## 2, DDMS的Dump View Hierarchy及 Hierarchy View

查看acivity内部的布局层次

# CPU

## 1, adb shell top –m n

显示前n个占用CPU最高的几个进程。

## 2， adb shell dumpsys cpuinfo

显示一段时间内cpu的使用情况。

## 3，DDMS的Threads和traceview

[**http://www.apkbus.com/android-131042-1-1.html**](http://www.apkbus.com/android-131042-1-1.html)

# Memory

## Dalvik对内存的限制

获取手机设置的内存的上限：adb shell getprop | grep dalvik.vm.heapgrowthlimit

**[dalvik.vm.heapstartsize]: [8m]** 启动一个应用程序之后为其分配的初始堆栈大小

**[dalvik.vm.heapgrowthlimit]: [128m]** 受控情况下的极限堆（仅仅针对dalvik堆，不包括native堆）大小，dvm heap是可增长的，但是正常情况下dvm heap的大小是不会超过dalvik.vm.heapgrowthlimit的值（非正常情况下面会详细说明）。这个值控制那些受控应用的极限堆大小，如果受控的应用dvm heap size超过该值，则将引发oom。

**[dalvik.vm.heapsize]: [512m]** 不受控情况下的极限堆大小，这个就是堆的最大值。不管它是不是受控的。这个值会影响非受控应用的dalvik heap size。一旦dalvik heap size超过这个值，直接引发oom。

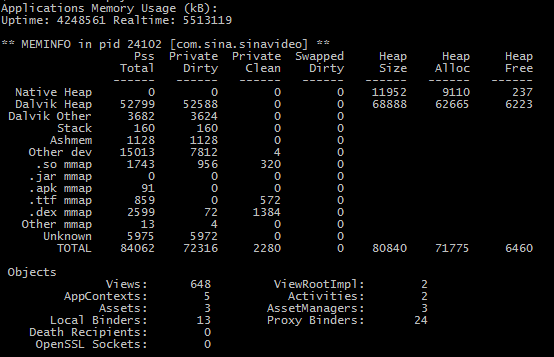
**[dalvik.vm.heapmaxfree]: [8m]**

**[dalvik.vm.heapminfree]: [2m]**

在android开发中，如果要使用大堆。需要在manifest中指定**android:largeHeap**为true。这样dvm heap最大可达dalvik.vm.heapsize。

## meminfo

获取内存的信息：adb shell dumpsys meminfo com.sina.sinavideo



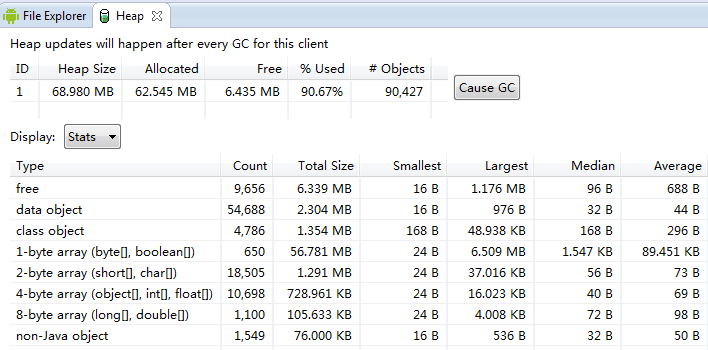
**Dalvik Heap Size：** Dalvik Heap总共的内存大小。

**Dalvik Heap Alloc：**Dalvik Heap分配的内存大小。

**Dalvik Heap Free：**Dalvik Heap剩余的内存大小。

Dalvik Heap Size 约等于Dalvik  Heap Alloc + Dalvik  Heap Free

## DDMS Update heap

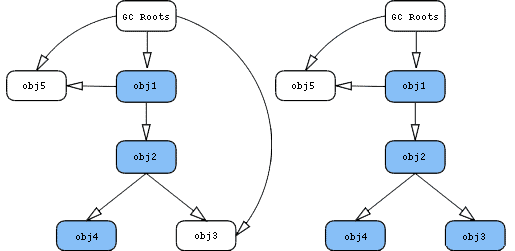


## DDMS Dump HPROF File

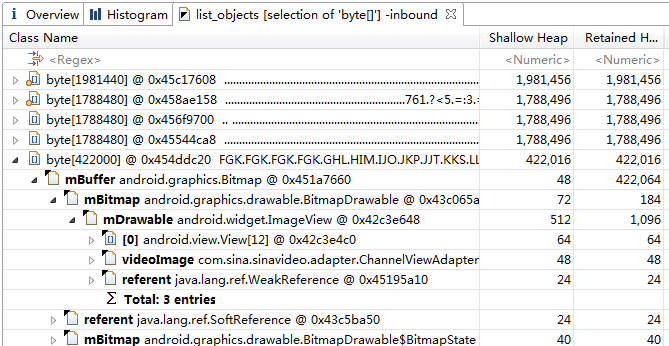
**Shallow size**就是对象本身占用内存的大小，不包含对其他对象的引用，也就是对象头加成员变量（不是成员变量的值）的总和。

**Retained size**是该对象自己的shallow size，加上从该对象能直接或间接访问到对象的shallow size之和。换句话说，retained size是该对象被GC之后所能回收到内存的总和。

左图，obj1的ratained size就是1,2,3。右图，obj1的retained size是1,2,3,4。



Histogram-->按shallow heap排序 -> List Objects -> with incoming references



## Procrank

**VSS** - Virtual Set Size 虚拟耗用内存（包含共享库占用的内存）

**RSS** - Resident Set Size 实际使用物理内存（包含共享库占用的内存）

**PSS** - Proportional Set Size 实际使用的物理内存（比例分配共享库占用的内存）

**USS** - Unique Set Size 进程独自占用的物理内存（不包含共享库占用的内存）