https://blog.csdn.net/yrwan95/article/details/82826066

**Eclipse Memory Analyzer —— MAT查找内存泄漏工具**

2018年09月24日 00:12:42 [蜜汁蛋总](https://me.csdn.net/yrwan95) 阅读数 821

 版权声明：转载请注明出处 https://blog.csdn.net/yrwan95/article/details/82826066

**一、MAT是什么？**

MAT(Memory Analyzer Tool)，一个基于Eclipse的内存分析工具，是一个快速、功能丰富的JAVA heap分析工具，它可以帮助我们查找内存泄漏和减少内存消耗。使用内存分析工具从众多的对象中进行分析，快速的计算出在内存中对象的占用大小，看看是谁阻止了垃圾收集器的回收工作，并可以通过报表直观的查看到可能造成这种结果的对象。

**二、如何使用MAT**

本文以一段小代码为例，讲述如何将dump文件导入MAT进行分析。

import java.util.ArrayList;

public class Test01 {

byte[] byteArray = new byte[1 \* 1024 \* 1024];*// 1MB*

public static void main(String[] args) {

ArrayList<Test01> list = new ArrayList<>();

try {

for (int i = 1; i <= 40; i++) {

list.add(new Test01());

}

} catch (Exception e) {

e.printStackTrace();

}

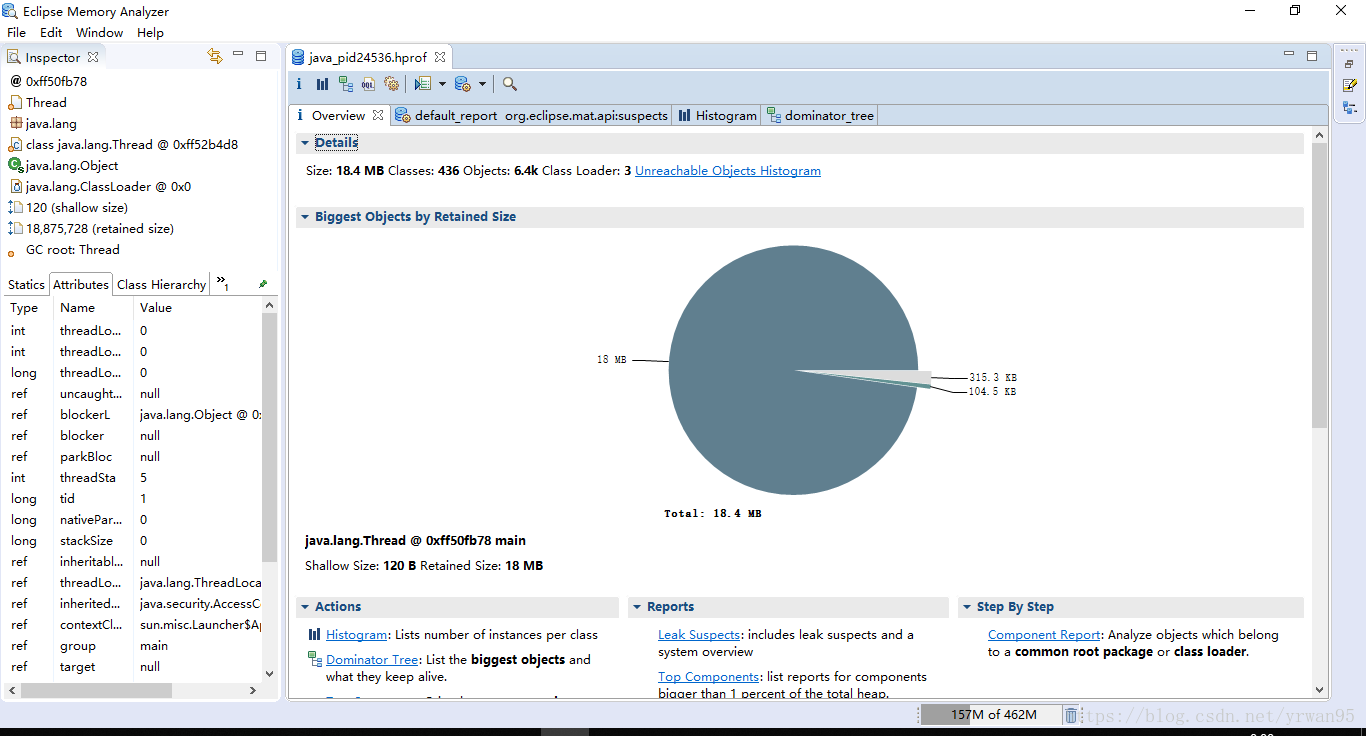
}

}

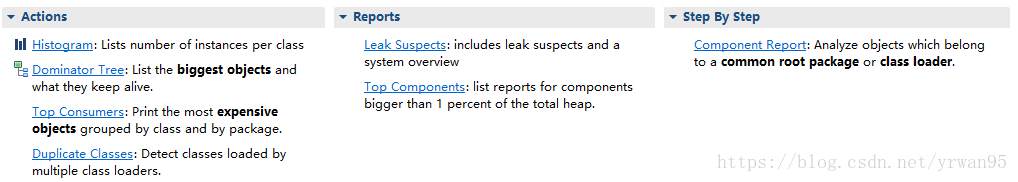
具体方法如下：

1. 通过 -XX:+HeapDumpOnOutOfMemoryError 参数来生成dump文件（为产生OOM，还使用了-Xms20m -Xmx20m参数）；
2. 使用Memory Analyzer打开生成的dump文件。“File->Open heap dump...”打开指定的dump文件后，将会生成Overview选项。

如下图：

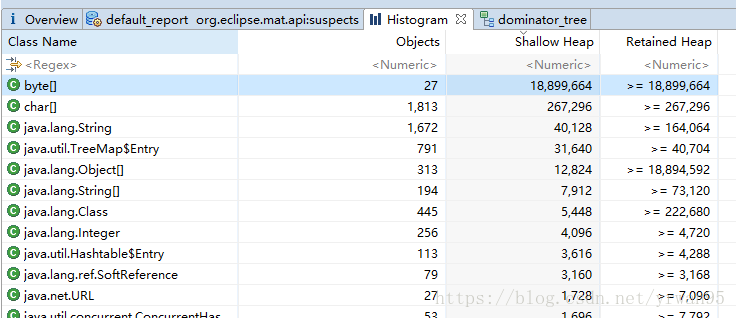


在Overview选项中，以饼状图的形式列举出了程序内存消耗的一些基本信息，其中每一种不同颜色的饼块都代表了不同比例的内存消耗情况。下方有几个选项如图：

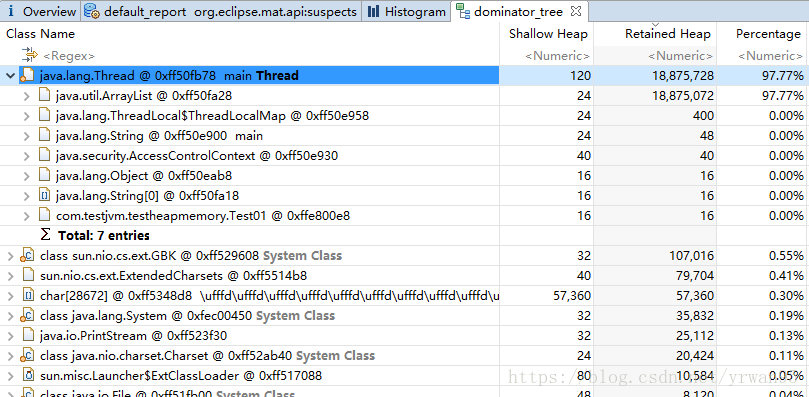


1. Histogram  可以列出内存中的对象，对象的个数以及大小；
2. Dominator Tree  可以列出那个线程，以及线程下面的那些对象占用的空间；
3. Top consumers  通过图形列出最大的object；
4. Leak Suspects  通过MA自动分析泄漏的原因。

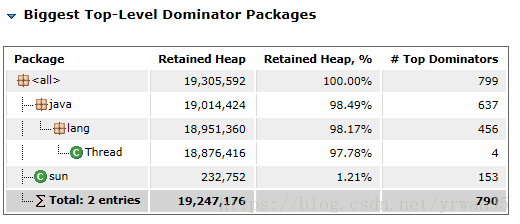
**Histogram：**



**Dominator Tree：**



**Top consumers：**



**Leak Suspects：**

