

Server side Memory Profiling



Memory Profiler

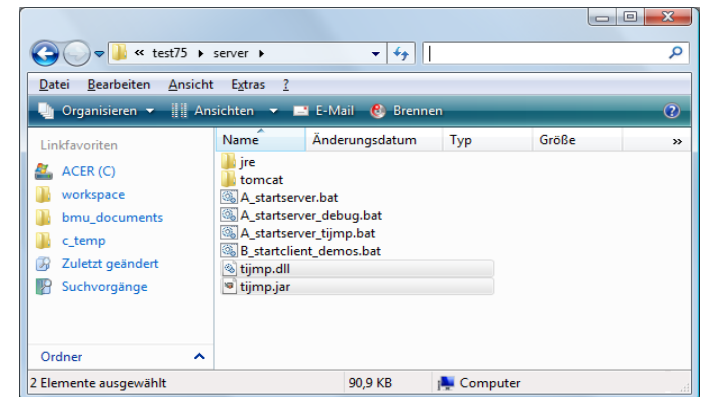
- In case of „constantly increasing memory“: any attempt of solving the issue without using a memory profiler is just not serious...!
- We internally use TIJMP
 - <http://www.khelekore.org/jmp/tijmp/>
 - Of course there are other profilers with more features available as well! TIJMP does the job we expect it to do, that's why we use it.

Installation

- Copy tijmp.dll and tijmp.jar into the default start directory for the Tomcat server of CaptainCasa
- Update the start.bat file for starting the Tomcat:

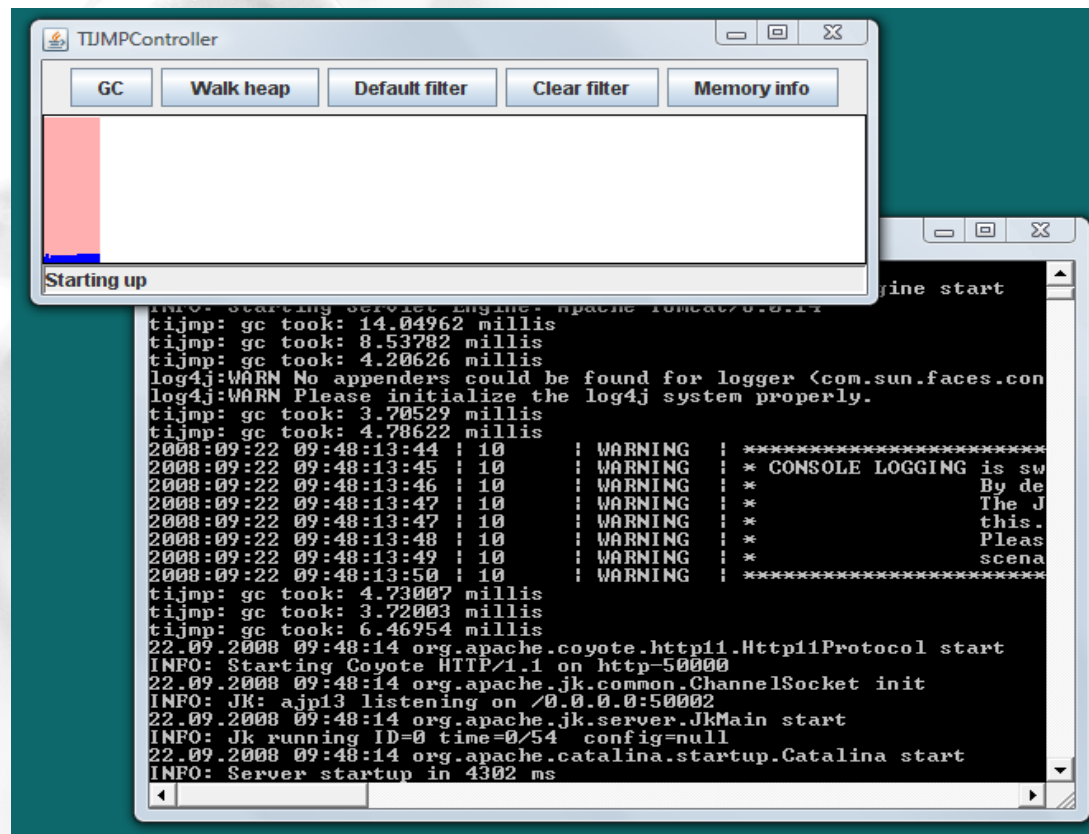
```
set JAVA_HOME=  
set JRE_HOME=jre  
set CATALINA_HOME=tomcat  
set JAVA_OPTS=-Xmx512m -Xms128m -agentlib:tijmp  
-Dtijmp.jar=tijmp.jar
```

```
tomcat/bin/catalina run
```



Result

- Tomcat is started, in parallel a profiler window is opened



Typical Steps

- Start Garbage Collection
- Start „Walk Heap“ in order to reset counters
- Do certain steps in your application, end the steps.
- Start Garbage Collection
- Start „Walk Heap“ again and analyze differences

The screenshot displays the CaptainCasa Enterprise Client interface. A window titled "Heap Walk Result 2" is open, showing a table with the following data:

Class	Count	Count ...	Size	Size diff
workplace.DemoHelloWorld	2	1	64 B	32 B
org.eclint.jsfserver.elements.impl.ROWDEMOB...	1	0	232 B	0 B
workplace.DemoIntro	1	0	48 B	0 B
workplace.DemoIntro\$ItemAction	3	0	48 B	0 B

The background shows a web application interface with a "Hello World" form and a "Hello!" button. The status bar at the bottom indicates "Hello World" and "72(26) ms".