

CCM JAVA Process Heap Dump and Garbage Collection Statistics

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Problem](#)

[Solution](#)

Introduction

This document describes how to add Java Virtual Machine (JVM) flags to the **JAVA_OPTS** in the CloudCenter Manager (CCM).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Linux Interface
- Virtual Machine Environment
- CCM component

Components Used

The information in this document is based on these software versions:

- CloudCenter version 4.x
- CCM appliance

Background Information

At times, the TOMCAT process goes unresponsive and stops to function properly but there is no information that indicates the reason why the TOMCAT went unresponsive.

Problem

Users are unable to log into CloudCenter and users who are logged in to the CloudCenter see the CCM User Interface (UI) unresponsive. This issue can be triggered because the TOMCAT service does not run and it has stopped to work. Then, the CCM default logs do not show any details or the reason as to why the TOMCAT service has stopped to work.

Solution

In order to collect additional information before the TOMCAT service stops to run, JVM flags need to be added to the **JAVA_OPTS** which is in the **/usr/local/tomcat/bin/setenv.sh**. These parameter helps you to collect the heap dump of the JAVA process and the Garbage Collection statistics of the JAVA process. Then, when the TOMCAT process becomes unresponsive, it generates **/usr/local/tomcat/logs/ccm.bin** and **/usr/local/tomcat/logs/ccm-gc.log**.

Step 1. Add these parameters at the **/usr/local/tomcat/bin/setenv.sh**.

```
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/usr/local/tomcat/logs/ccm.bin -verbose:gc -
XX:+PrintGCDetails
-XX:+PrintGCDateStamps -Xloggc:/usr/local/tomcat/logs/ccm-gc.log
```

Step 2. Navigate to **CCM > /usr/local/tomcat/bin/setenv.sh**

```
[root@ccm bin]# cat /usr/local/tomcat/bin/setenv.sh
export LD_LIBRARY_PATH=/usr/local/apr/lib
export CATALINA_PID=${CATALINA_HOME}/catalina.pid
export JPDA_ADDRESS=8000
export JAVA_OPTS="-Djava.awt.headless=true -Dfile.encoding=UTF-8 -server -Xms1024m -Xmx2048m -
XX:NewSize=512m
-XX:MaxNewSize=512m -XX:PermSize=512m -XX:MaxPermSize=512m -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=/usr/local/tomcat/logs/ccm.bin -verbose:gc -XX:+PrintGCDetails -
XX:+PrintGCDateStamps
-Xloggc:/usr/local/tomcat/logs/ccm-gc.log -Dhttp.proxyHost=rcdn-calo-proxy.cisco.com -
Dhttp.proxyPort=3128
-Dhttp.nonProxyHosts=10.201.210.1" export CLASSPATH=/usr/local/osmosix/etc export
CATALINA_OUT=/dev/null
```

Step 3. Restart the CCM TOMCAT process.

```
/etc/init.d/tomcat restart
```

Step 4. After the restart, execute the command **ps -ef | grep tomcat** in order to confirm that the TOMCAT process uses the new parameters that were added to the **/usr/local/tomcat/bin/setenv.sh**.

```
[root@ ccm ~]# ps -ef | grep tomcat
cliqrus+ 18683      1 99 16:46 ?          00:00:27 /usr/lib/jvm/java-8-sun/bin/java
-Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -
Dfile.encoding=UTF-8
-server -Xms1024m -Xmx2048m -XX:NewSize=512m -XX:MaxNewSize=512m -XX:PermSize=512m -
XX:MaxPermSize=512m
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/usr/local/tomcat/logs/ccm.bin -verbose:gc -
XX:+PrintGCDetails
-XX:+PrintGCDateStamps -Xloggc:/usr/local/tomcat/logs/ccm-gc.log -Dhttp.proxyHost=rcdn-calo-
proxy.cisco.com
-Dhttp.proxyPort=3128 -Dhttp.nonProxyHosts=10.201.210.1 -
Djava.endorsed.dirs=/usr/local/tomcat/endorsed
```

```
-classpath  
/usr/local/osmosix/etc:/usr/local/tomcat/bin/bootstrap.jar:/usr/local/tomcat/bin/tomcat-juli.jar  
-Dcatalina.base=/usr/local/tomcat -Dcatalina.home=/usr/local/tomcat -  
Djava.io.tmpdir=/usr/local/tomcat/temp  
org.apache.catalina.startup.Bootstrap start root 18765 17947 0 16:47 pts/1 00:00:00 grep --  
color=auto tomcat
```

These new parameters helps you to create additional files that can be collected once the TOMCAT process goes unresponsive again:

- Heap dump of the JAVA process - **/usr/local/tomcat/logs/ccm.bin**
- Garbage collection stats of the JAVA process - **/usr/local/tomcat/logs/ccm-gc.log**