
Java™
ORACLE

Java Mission Control & Java Flight Recorder

Java Your Next (Cloud)

Vaibhav Choudhary (@vaibhav_c)
Java Platforms Team
https://blogs.oracle.com/vaibhav

Agenda of the day …

1. Understanding Java Mission Control
2. Understanding Java Flight Recorder
3. Data Collection and Analysis by JMC and JFR
4. Incident Analysis and Demo

“Once again, @javamissionctrl is saving my day.”
– Michael Nitschinger

“Mission Control is what you wanted instead of profiler, but were afraid to ask.”
– Oleksandr Otenko, Oracle Performance Engineer

“JMC, because in my experience other tools cause way more harm than good.”
– David Buck, Oracle SE

“It’s an invaluable piece of work.”
– Maurizio Cimadamore, Oracle (Java Lang Tools)

“If I could get only one present for Christmas, it would be Oracle Java Mission Control.”
– Morten G. Hermansen (tweet)

“JFR is really impressive for partners in Japan. Members who have trouble shooting tasks totally understand the necessity of JFR.”
– Tomofumi Nijo, Oracle (Japan, Sales Consulting)

What is Java Mission Control ?

- Set of powerful tools to monitor and manager Java Application.
- User friendly JMX console
- Can analyze Memory, Garbage Collector, Code and many more
- Newer version can help you with Automated Analysis
- You can connect a live application OR you can analyze a recorded application (by JFR).
- Designed to be very low overhead (less than 2%)

What is Java Flight Recorder ?

- Just like “Black Box” in aircraft
- A profiling and event collection Framework
- Gather low level information from JVM without changing the runtime behavior of application
- Can be used (Should be used) in production. Very less overhead
- JFR recordings can be analyzed by JMC

Enabling JFR

- VM need to start with following options
  - java -XX:+UnlockCommercialFeatures -XX:+FlightRecorder
- Different options like
  - java -XX:+UnlockCommercialFeatures -XX:+FlightRecorder -XX:StartFlightRecording=duration=60s,filename=myrecording.jfr HelloWorld
  - java -XX:FlightRecorderOptions=defaultrecording=true,dumponexit=true,dumponexitpath=/tmp/file.jfr
  - We can use jcmd (available in JDK/bin) as well
    - jcmd PID JFR.start duration=60s filename=file.jfr

Running JMC

- JAVA_HOME/bin/jmc