



JMC - JDK Mission Control

JMC - JDK Mission Control

Overview

Definition from oracle.com :

JDK Mission Control (JMC) is an advanced set of tools for managing, monitoring, profiling, and troubleshooting Java applications.

JMC enables efficient and detailed data analysis for areas such as code performance, memory, and latency without introducing the performance overhead normally associated with profiling and monitoring tools.

JDK Mission Control is a tool made to be used in conjunction with *JFR tool*.

Installation (Windows)

Download JMC from [oracle website](#).

Extract the contents of the archive under the directory of your choice (`$install_dir`).

JMC is available as an .exe at `$install_dir/JDK Mission Control/jmc.exe`

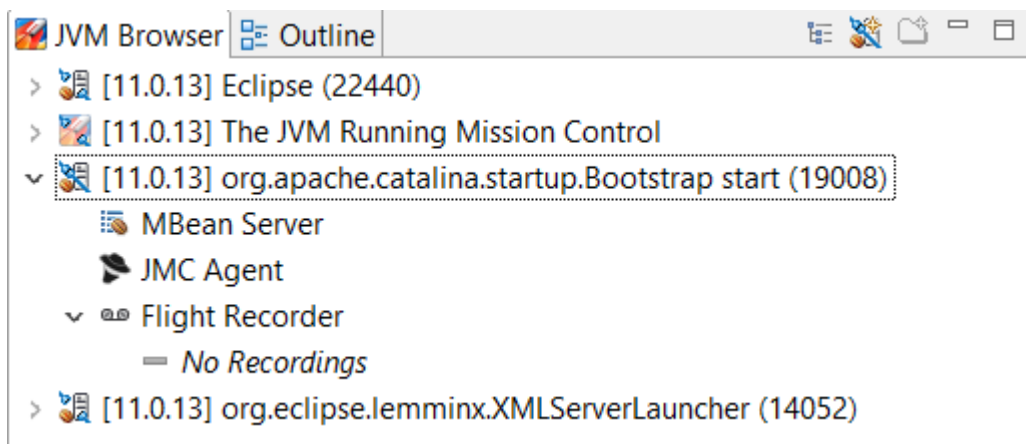
Use

When you start the JMC application for the first time, a number of views open by default.

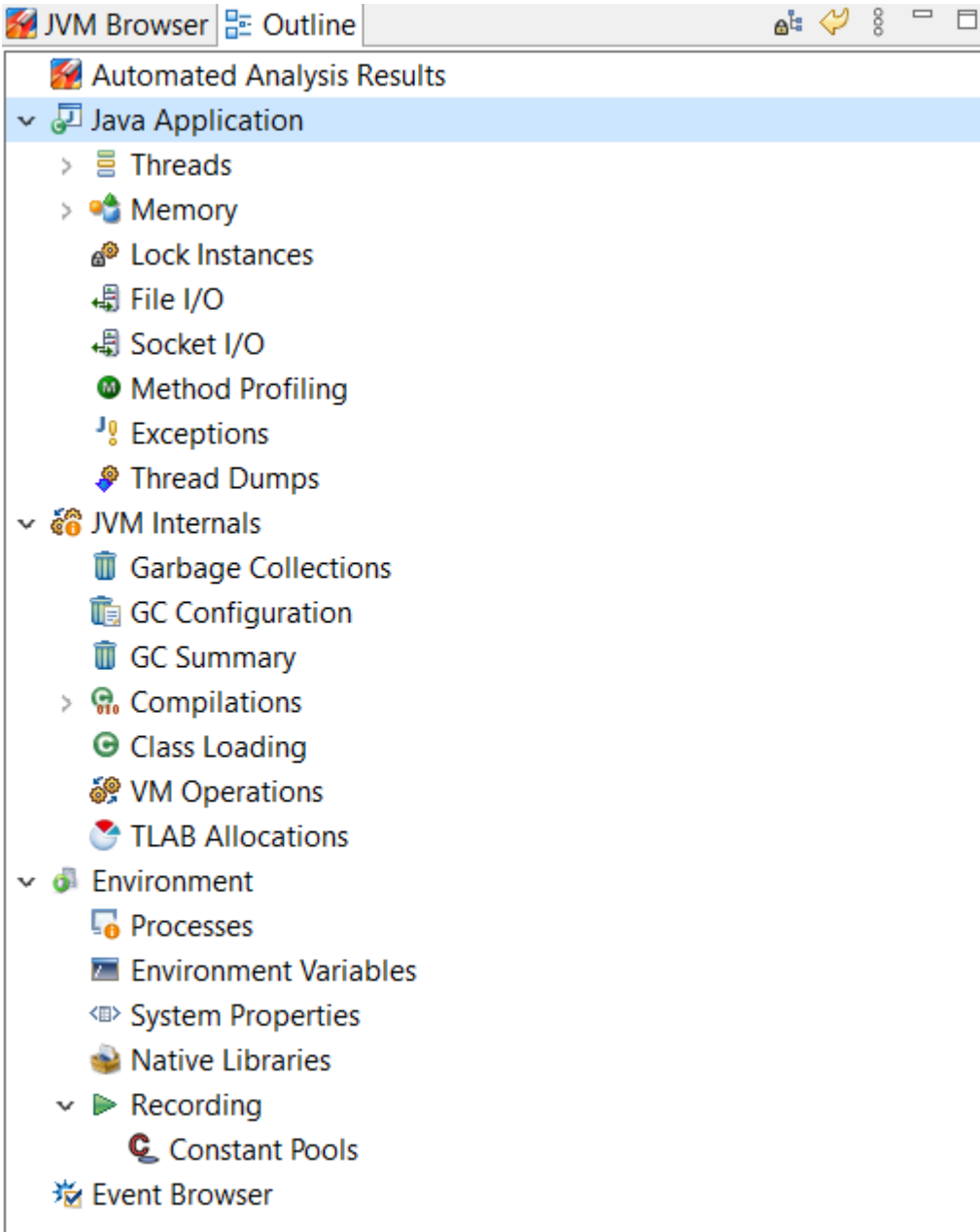
You can open the Window menu, select Show View and then Other to open a window that lists all available views. Select the necessary views and click OK to open them. The following views are available.

- JVM Browser: Lists all the JVM instances running locally (on the host) and JVMs discovered on the network.

The JVM Browser can be viewed in two different modes: as a flat list, and as a tree (visible by default).



- Outline: Shows the data collected in a Flight Recording. It organizes and presents flight recording data as pages in a tree for easy navigation (visible by default).



- Progress View: Displays the progress of running operations, for example, a flight recording.



- Properties: Lists the properties of items that you select in tables, including hidden properties that are not displayed in the tables (visible by default).

Field	Value
Event Type	Too many values...
Start Time	2022-09-30 15:25:08.000 - 15:26:08
Duration	0 s - 52.325 s
End Time	2022-09-30 15:25:08.000 - 15:26:08
Event Thread	Too many values... 22,342 events

- Results: Displays a list of rules with their corresponding scores (visible by default).

It also shows the results from the automated analysis relevant to the currently opened page in the editor.

Count	Rule Name
20	Threads Allocating
1	Method Profiling
5	Application Halts
0	High JVM CPU Load
0	Parallel Threads

- Stack Trace: Displays stack traces for the recorded events (visible by default).

Stack Trace	Count	Percentage	Percentage (By Duration)
void java.lang.Thread.run()	13819	77.8 %	30.9 %
void com.lmax.disruptor.BatchEventProcessor.run()	5183	29.2 %	1.94 %
void com.lmax.disruptor.BatchEventProcessor.processEvents()	5183	29.2 %	1.94 %
long com.lmax.disruptor.ProcessingSequenceBarrier.waitFor(long)	5178	29.1 %	1.94 %
long com.lmax.disruptor.TimeoutBlockingWaitStrategy.waitFor(long, Sequence, Sequence, Se...	5178	29.1 %	1.94 %
long java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject.awaitNanos(lon...	5178	29.1 %	1.94 %
void java.util.concurrent.locks.LockSupport.parkNanos(Object, long)	5129	28.9 %	1.94 %
void jdk.internal.misc.Unsafe.park(boolean, long)	5129	28.9 %	1.94 %

Note

You can rearrange these views by dragging and dropping them to the desired location. You can also maximize and minimize the views.