

ORACLE JAVA SE ADVANCED

ENTERPRISE GRADE JAVA SE MONITORING AND DIAGNOSTICS

KEY FEATURES

- Latency analysis
- Automatic deadlock detection
- Memory leak detection and identification
- Zero performance overhead
- After the fact diagnostics
- Garbage collection visualization
- Usage tracking

KEY BENEFITS

- Always on, Java level, detailed diagnostics
- Designed for production use, the tools do not interrupt your system in any way
- Monitor key system values and diagnose root causes in minutes
- Keep track of all your Java installations, and control upgrade schedules

Oracle Java SE Advanced is a package designed to help enterprises that use Java SE as a part of their mission critical infrastructure. With features to handle in-production diagnostics and monitoring, enterprise wide usage tracking and after-the-fact incident logging Java SE Advanced is uniquely suited to assist the modern enterprise.

Java Mission Control

Finding intermittent issues in your production system requires you to enable performance logging diagnostics frameworks or tools. Obviously you cannot afford running those in production all the time. So every time a problem occurs, in hindsight, you realize that you should have enabled the heavy logging at the last restart.



Java Mission Control solves the diagnostics problem in multiple ways.

- Always enabled. No restarts needed.
- Zero performance overhead. You can have it running all the time
- No libraries to include, no code changes. The diagnostics features are included in the JVM so you do not have to do anything.
- Flight Recorder feature. Always on circular buffer recording allows you to record an issue after the fact.

The Java Virtual Machine (JVM) performs a number of functions behind the scenes. Things like garbage collection, thread scheduling and automatic code optimization. The data needed for the JVM to perform this work is mostly hidden to the user and programmer. Java Mission Control brings forth this hidden data and combines it with user-friendly visual tooling to provide deep insight into all the inner workings of the JVM.

The server part of Java Mission Control is built into the JVM, giving it access to JVM internals at no performance cost to the application. Not only does it have the advantage of zero

overhead and the fact that the developer does not have to adapt his code in any way, it also supplies information that used to be impossible to get.

Example of use:

- Diagnose and identify where in the code there is a deadlock in less than 10 seconds
- Figure out exactly how much time is spent in each method
- Visualize all sources of latency, including where in the source code they occur.

Java Mission Control and the Flight Recorder feature is currently fully supported on JRockit and partially supported on Hotspot.

Flight Recorder

Two hours past midnight on a Sunday your system stops responding to customer requests. Your IT staff on call does the only thing they can in the middle of the night on a Sunday– they reboot the servers. Now, if you had detailed logging turned on you would have been able to find out why the problem occurred but having that level of logging leads to performance issues so you cannot have it on all the time.

Sounds familiar? If not congratulations, but for the rest of us it describes a familiar nightmare scenario with very few good solutions. Java Mission Control would give you the details you needed, but you were not recording at the time.

This is why we created the Flight Recorder feature. Simply put – it is a continuous Java Mission Control recording of your system. Always on in a circular, last in last out, buffer. This allows you to do after the fact analysis of intermittent system issues, SLA breaches and crashes. With Flight Recorder you can see what actually led up to the issue, not just what happened after.

Usage Tracking and Auto Update Off

Do you know exactly what versions of Java are running in your organization? Are all the startup flags optimized on all systems for running your applications? If not, have you done quality assurance on all possible versions with all possible flags?

The Usage Tracking feature collects information about the Java environment and sends it to a specified server or saves it to disk, giving you complete visibility into what is running, where it's running, on what and how.

Example of data that is being logged by the Usage Tracker:

- Java and JVM versions
- Java application name
- Host name and IP address
- Date and time
- JVM arguments
- Class path

To support the visibility given by the Usage Tracker, we also supply a version of the JVM with auto update turned permanently off. This gives IT complete control of when to change Java versions within the company.

Contact Us

For more information about [insert product name], visit oracle.com or call +1.800.ORACLE1 to speak to an Oracle representative.



Copyright © 2012, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0112

Hardware and Software, Engineered to Work Together