

MERCURY DIAGNOSTICS PROFILER

Mercury Diagnostics Profiler™ for J2EE is a lightweight, no-cost application performance profiler for Java and J2EE developers. It uses the same byte code instrumentation probe as the Mercury Diagnostics™ family, enabling a single-vendor lifecycle solution.

Are you a JAVA developer under pressure to resolve performance problems earlier in the application lifecycle? Do your responsibilities include the configuration and management of J2EE application server environments in testing and production? Do you use several different tools to analyze and diagnose application development problems?

Available as a no-cost download, Mercury Diagnostics Profiler features the capabilities most sought after by application owners in solving tough application problems. It enables you to collect critical diagnostics information to make applications ready for load testing and to solve problems earlier in the application lifecycle.

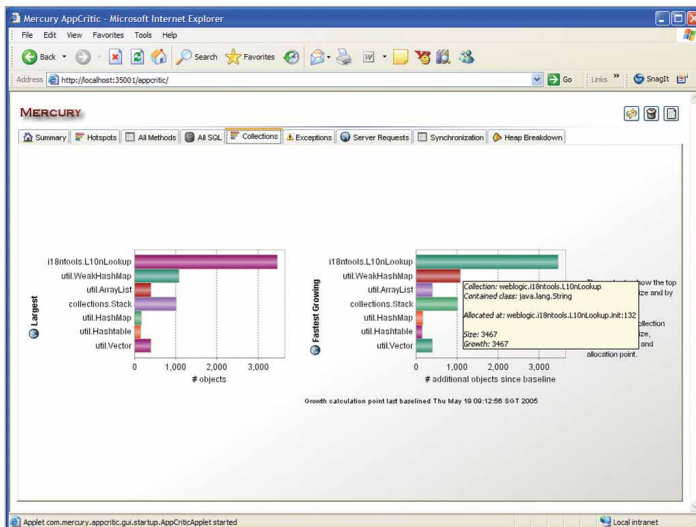
Unlike existing tools for developers, Diagnostics Profiler integrates into the lifecycle of load testing and problem isolation in production by sharing the same lifecycle probe as Mercury Diagnostics – our diagnostics solution for monitoring, triaging, and diagnosing application performance problems in testing and production environments. It enables you to diagnose application performance problems across the performance management lifecycle.

Developers currently employ multiple diagnostics tools – from built-in JVM features like “verbosegc” to costly third-party vendor tools – resulting in disparate data for analysis. In addition, existing tools are often unable to capture diagnostics information for multiple users, making diagnosing problems across the lifecycle very difficult.

Diagnostics Profiler is part of the Mercury Diagnostics product family, which is a single-vendor lifecycle performance diagnostics solution that eliminates the need to use several different tools to analyze and diagnose application development problems. In addition, it's simple to install and provides immediate value with diagnostics information that helps you quickly resolve performance problems.

How it Works

The Diagnostics Profiler probe uses byte code instrumentation, enabling your application to capture diagnostics data that is relevant to end users. It provides concrete information that points to the root cause of problems, arming you with actionable information. When installed, the probe captures appropriate diagnostics data



Lightweight memory diagnostics in JAVA collections.

and sends it to the profiler data sink. The applet-based diagnostics UI presents this data to the user in ways that can help you solve the most common application problems found in the application lifecycle.

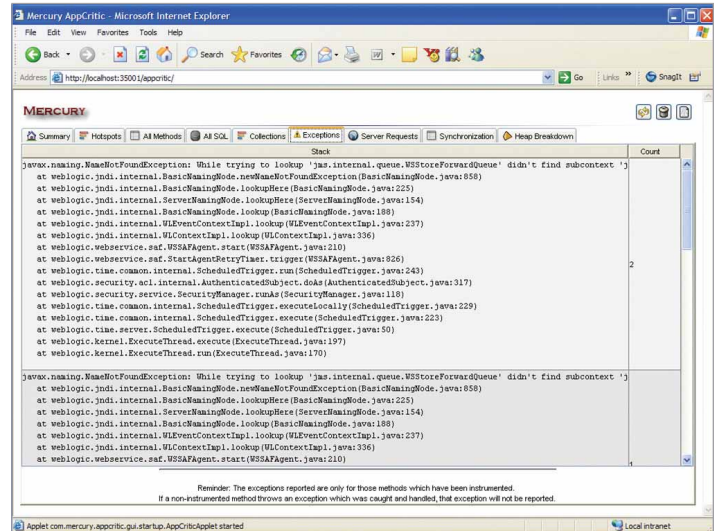
When you release your application to QA for load testing, QA engineers can use Mercury LoadRunner Diagnostics (not included in download) by leveraging the same lifecycle diagnostics probe to triage and identify problems under load conditions. Then, when you deploy your application in production, your operations and support teams can use Mercury Diagnostics (not included in download) to leverage the same lifecycle diagnostics probe to collect aggregated data on a 24x7 basis for application monitoring.

Key Functions

Memory Diagnostics allows you to quickly identify the top fastest-growing and largest collections, including the size of the collection and where in your code the collection was allocated. Mercury Diagnostics Profiler provides memory diagnostics in JAVA collections. It also allows you to take heap snapshots to obtain byte size for objects.

Exception Diagnostics allows you to quickly determine if your application is throwing exceptions and exactly what those exceptions are. Often, exceptions are “eaten” without good reason, causing significant problems under load. This feature enables you to quickly identify problem areas so you can fix them before you release to QA.

Heap Breakdown allows you to quickly get a dump of your heap to find out if and exactly which objects are leaking. This provides you with the ability to force a Garbage Collection (GC) and capture snapshots on demand or automatically when heap growth crosses a threshold.



Exception stack traces.

FEATURES AND BENEFITS

- **Summary and hotspots:** Identify top performance hotspots in applications.
- **Server request breakdown:** Identify where time is spent.
- **Layer breakdown:** Identify which J2EE layer is slow.
- **Slowest roots:** Identify slowest server request or application entry points for non-web-fronted applications.
- **Top three slowest instances:** Identify outliers for intermittent problems.
- **VM heap usage:** Identify memory problems/GC.
- **Collection memory leak diagnostics:** Identify fastest-growing and largest-size JAVA collections including the caller and the exact line number where collection was allocated.
- **Heap breakdown with class and size information:** Identify leaking object, growth trends, instance count, and byte size for objects.
- **SQL diagnostics (slowest SQL):** Identify slowest SQL query with query information.
- **Synchronization diagnostics:** Identify locks with hold times.
- **Exception diagnostics (including exception traces and count):** Identify exception counts and trace information that often might go undetected.

MERCURY™

Mercury is the global leader in business technology optimization (BTO). We are committed to helping customers optimize the business value of IT.
WWW.MERCURY.COM