



# The Mainstream

An article from the IBM @server zSeries software newsletter

Find the current version of The Mainstream here [ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

## J2EE Management Made Easy: IBM WebSphere Studio Application Monitor

There are several great reasons to love Java® 2 Enterprise Edition (J2EE). This platform for large server applications helps today's enterprises quickly develop and deploy multi-tiered applications. In today's heterogeneous, multi-vendor applications environment, J2EE provides a single standard that can sit atop existing enterprise systems — database management systems, transaction monitors, naming and directory services, and more.

Breaking down the barriers between these systems, J2EE enables developers to build the next generation of components, tools, systems and applications. It provides a rich environment for defining business logic. It delivers a wide array of highly available and scalable technologies that link back-end systems to middle-tier business logic. And it can interact with virtually any computing resource, including databases, enterprise information systems, portals, network components and desktops.

All that advanced functionality, however, comes at a price: a level of complexity that can pose difficult management problems for even the most sophisticated IT shops. All resources related to the J2EE application server must be integrated to enable applications to run, and must scale in performance as application demand increases. That's no small task.

And that's why so many J2EE-based organizations are turning to IBM WebSphere® Studio Application Monitor (WSAM), a powerful management solution that helps application developers, testers and analysts monitor application performance, availability and related environmental factors.

With WSAM, you can construct, troubleshoot and manage your J2EE applications. Since e-business applications operate within the boundaries of a J2EE application server, all elements — servlets, Java Beans and Enterprise JavaBeans — can be well defined and managed. WSAM:

- Collects data about application and system performance



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

- Provides a comprehensive architecture for analyzing information about them
- Displays that information on a standard web browser, without requiring users to install client code

### **Opening up the Java ‘black box’**

All these functions are integrated in a single product that manages multiple platforms, including mixed-vendor application server environments such as mainframe-based J2EE Server and WebSphere Application Server for z/OS®. This means that you can perform in-depth problem determination and performance analysis of mission-critical J2EE applications that are running on the WebSphere Application Server platform. WSAM lets you open up the Java “black box” to see what’s happening within your applications, then track and trend performance and resource consumption patterns over time.

### **Analysis and reporting, both high-level and granular**

WSAM provides a variety of analytical and reporting capabilities, from high-level enterprise views down to specific thread and method-level details and traces. It captures and stores detailed resource utilization data at a transactional level, and can display application and server activity both historically and in realtime. This granular historical data can be used to:

- Make application sizing and capacity planning decisions
- Anticipate and monitor peak resource demands
- Consolidate workloads to adhere to enterprise application Service Level Agreements

WSAM functions entirely independent of J2EE applications, regardless of whether they were developed in-house or by a third-party developer. Without requiring changes in application code, it can monitor heterogeneous environments — such as WebSphere Application Server on z/OS and on distributed platforms — without resorting to instrumentation or other modifications to the application server. That makes WSAM very easy to implement.

[ibm.com/software/  
zseries/mainstream](http://ibm.com/software/zseries/mainstream)

### Identifying problems before they occur

Every day, you encounter a daunting array of application and environmental factors that can degrade an application's performance, and even cause outages. When you face excessive service request times, it's usually due to factors such as:

- Application server configuration, capacity and clustering issues
- Poor application design (bugs, memory leaks, code inefficiencies)
- Large or excessive HTTP sessions
- Insufficient resource pools (thread, db connection, EJB)
- Sub-optimal database design
- Java Virtual Machine (JVM) problems (heap, garbage collection, incorrect parameters)
- Network congestion, including firewall, proxy server and load balancer
- Hardware issues and outages

Identifying these factors before they cause performance problems is critical to preventing slow-downs and reducing calls for network support. And if problems do happen, it's important to have a solution that allows rapid and deep visibility into application behavior, to minimize or eliminate downtime.

This is the province of WebSphere Studio Application Monitor. Its top-down approach manages total availability from an application perspective. Using the standard programming model built into J2EE and JVM, data can be captured from inside the application server, for effective performance management, exception handling and capacity planning.

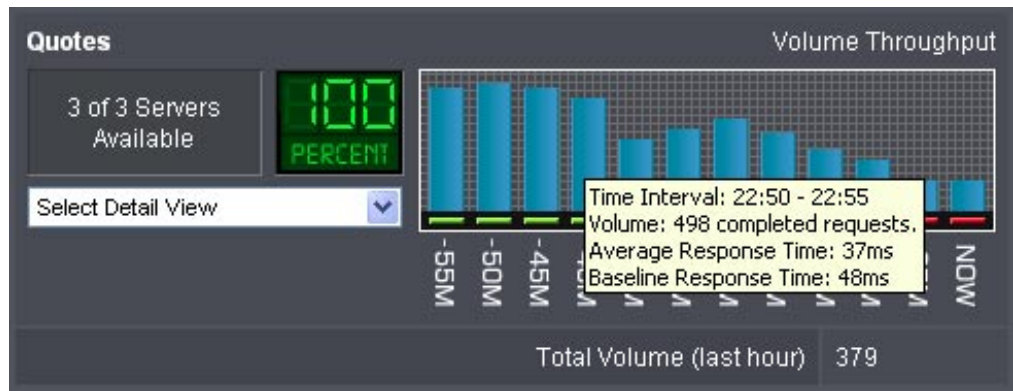
### A dashboard for quickly spotting problems

With WebSphere Studio Application Monitor, you can easily spot problem conditions, via a “dashboard” that displays enterprise-wide view J2EE application and server availability and shows all applications running in assigned server groups.

The key to successfully identifying problems: quickly discovering and pursuing conditions based on symptoms and their surrounding context — for example, a sudden drop in throughput accompanied by an increase in heap size or garbage collection frequency.

[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

WSAM makes it easy to check these anomalies as they occur. It captures application behavior in realtime to rapidly isolate problem sources, then approaches the problem based on specific, well-defined characteristics. This is especially valuable in complex, multi-tiered distributed environments, where it can be difficult to identify and manage problems associated with object-oriented applications.



*This WSAM dashboard, which displays a group of servers running a Quotes application, shows overall availability, performance and throughput of Quotes in five-minute increments over an hour. Red bars indicate a potential problem, since response times exceed a specified threshold while throughput levels are steadily decreasing. This condition can be investigated by using WSAM to analyze the J2EE environment.*

## Drill down with precision

WSAM features a well-structured and easy-to-navigate user interface that lets you drill down to identify and resolve application issues wherever they may occur. The interface connects you with WSAM's powerful capabilities, which include:

*In-Flight Request Search.* This unique feature quickly spots problems by immediately displaying user requests and their associated values (such as host server, resident time and thread ID). That makes it easy to spot candidates that are executing inefficiently in the system, and quickly obtain further information about offending requests.

Once identified, drill-down features can reveal a rich, detailed set of request metrics, providing a complete picture of what's going on inside the request itself. If it's not immediately obvious where



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

the problem lies, you can drill down further, to the request's session object, stack or even method trace data. You can even change the thread status or cancel a request altogether to free up system resources.

*Monitoring On Demand.* This technology provides three different monitoring levels for each managed server:

- *Level 1* monitors server and system resource availability in realtime, and captures application performance metrics for workload characterization with very low overhead.
- *Problem Determination Level 2* provides deeper focus on J2EE Application Programming Interfaces (APIs), including EJBs, to characterize relationships between applications and common business objects and services.
- *Profiling Tracing Level 3* provides full trace capability down to the method level, and automatically adjusts based on pre-set thresholds and timing parameters.

Once you've determined that an application has "hot spots" of degraded performance or excessive resource utilization, you can capture actual method entry and exit points. A complete trace will usually identify exactly where the problem is located. Method trace data can then be saved to an Adobe PDF file and shared with an application developer for analysis.

*Performance Analysis and Reporting.* WSAM helps track and trend historic application data. This is an important factor in understanding application usage patterns and associated systems resource requirements, or in planning for future capacity upgrades to handle increasing demand. It provides breakdowns of multiple combinations of performance statistics, measured by time, server group, request type, application and other variables. Also, it easily calculates charge-backs to other departments or customers who use applications running in a J2EE environment.

### **Pinpoint the root causes of problems**

Identifying transient or infrequent problems is often difficult, because their root causes can be subtle and involve complex relationships among conditions that vary over time. In these cases, it's essential to capture actionable data about system and application resources in order to understand the root-level cause of a problem.



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

If you can pinpoint a problem but can't reproduce it, WSAM's Traps & Alerts feature lets you dynamically capture a "snapshot" of the J2EE system at the time the event occurs, based on parameters you define. You can then configure specific actions to be triggered when those parameters are exceeded, and even set multiple thresholds for the same event. Once captured, alerts can be passed via email or Simple Network Management Protocol (SNMP), or saved to a searchable alerts repository.

You can set trap parameters based on any number of conditions, such as a specific application request, method, SQL call, and other erratic conditions such as excessive resident, CPU, or wait time. For example, if an application is consuming excessive CPU time, you can easily identify specific methods that consume CPU.

### **Talkin' trash...and garbage collection**

Heap memory is typically shared among all applications running within a JVM. And demand for that memory is affected by the number of requests generated by those applications. To reclaim "heap memory" that's allocated to inactive objects and unused threads, JVM performs Garbage Collection (GC), which often involves a trade-off between GC frequency and the duration of each collection.

WebSphere Studio Application Monitor stays on top of JVM Garbage Collection by showing a comprehensive set of GC attributes, including:

- The relationship of GC frequency to JVM CPU utilization
- Transaction response times and throughput affected by GC frequency and duration
- GC frequency and duration impacting heap size
- Heap usage as a contribution to GC duration
- The relationship of GC duration to system paging rate
- Average heap size after GC
- Number of GC occurrences and total GC elapsed time
- Heap usage as related to server workload, throughput volume and total users
- Uncollected memory values over time compared to GC cycles and paging rates

[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

- Uncollected memory related to changes in total users and throughput and/or response times
- Average response times related to number of active sessions and total requests

### Quickly and easily identify elusive memory issues

The Java environment handles memory management for its applications — which means that tracking down memory issues within an application can be difficult.

WSAM handles it. Using its powerful Memory Diagnosis capability, you can determine hard-to-find memory issues without resorting to tedious trial-and-error methods (such as removing and replacing portions of application code to localize a problem). Developers or operators can generate real-time reports about system memory indicators over time by plotting heap utilization over hourly intervals.



*If a report shows us that heap size is growing, it's easy to correlate the trend with other critical metrics, such as Garbage Collection frequency. In this example, heap size is steadily increasing despite frequent GCs, suggesting a memory leak.*



[ibm.com/software/zseries/mainstream](http://ibm.com/software/zseries/mainstream)

## The new frontier in J2EE application management

If you're an IT manager in today's complex, distributed, multi-tier J2EE environments, you've got to maintain highly available and scalable infrastructures for running mission-critical enterprise applications. Getting a handle on problem hot spots — and efficiently diagnosing and remedying potentially hard-to-find application problems — requires an application management solution that's quick to implement, delivers immediate value and scales to meet your needs as they grow.

Websphere Studio Application Monitory is that solution — a proven technology that represents a new frontier in J2EE application management. WSAM is constantly evolving to manage non-J2EE

---

### WSAM at a glance...

---

Effectively manage your entire J2EE environment.

---

Manage and monitor J2EE applications, from development to testing and deployment.

---

Diagnose application problems in realtime, rather than reviewing aggregated statistics that don't pinpoint where problems occur.

---

Capture and analyze problem data in realtime.

---

Automatically gather relevant data when problem conditions occur, and send alerts via email or SNMP.

---

Diagnose the root cause of memory leaks and other issues, such as long-lived objects in memory.

---

Easily change the depth of data based on pre-defined thresholds and time intervals.

---

Fix intermittent application problems caused by out-of-synch servers, including installed binaries and runtime environments.

---

Manage and monitor applications from any standard web browser.

---

transactional systems, such as CICS®, for end-to-end systems visibility.

If you're deploying J2EE infrastructure or evaluating management and monitoring solutions, take a close look at the unique capabilities of WSAM. You'll soon understand why it's in a class by itself.

If you are interested in learning more about WSAM, or would like to evaluate the product, please contact your local IBM Account Executive for more information.





## The Mainstream

9

[ibm.com/software/  
zseries/mainstream](http://ibm.com/software/zseries/mainstream)

IBM, CICS, WebSphere and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java, J2EE and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

References to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.