

*IBM Z OMEGAMON for JVM Reference
Guide*



Contents

Reference..... 1

- User interfaces..... 1
 - The OMEGAMON enhanced 3270 user interface..... 1
 - The Tivoli Enterprise Portal..... 1
- User interface help reference..... 2
 - Workspaces..... 3
 - Attributes..... 13
 - Situations..... 25

Index..... 29

Reference

This reference material provides information about the supported user interfaces in IBM® OMEGAMON® for JVM (IBM Z OMEGAMON for JVM) and also about monitoring resources that you can view in the user interfaces to suit your specific requirements.

User interfaces

IBM Z OMEGAMON for JVM supports z/OS-based OMEGAMON enhanced 3270 user interface as well the graphical Tivoli Enterprise Portal interface, which is based on distributed platforms.

The IBM® Tivoli® OMEGAMON enhanced 3270 user interface requires that you install one or more Tivoli Enterprise Monitoring Servers, but it offers plex-level as well as system-level and cross-product data and enhanced functionality. The Tivoli Enterprise Portal interface requires that you install one or more Tivoli Enterprise Monitoring Servers and a Tivoli Enterprise Portal Server, which must be installed on a distributed platform.

The OMEGAMON enhanced 3270 user interface

The OMEGAMON® enhanced 3270 user interface (enhanced 3270UI) retains the virtues of the earlier 3270 interfaces while it offers many of the features of the Tivoli Enterprise Portal in a native 3270 interface.

The OMEGAMON® enhanced 3270 user interface offers the following features:

- The ability to view plex-wide data, data from multiple systems, and data from multiple agents in the same interface or even the same workspace
- The ability to modify product-provided workspaces to create customized views and queries, and or create new workspaces
- The ability to set dynamic filters
- The ability to define or modify thresholds that trigger status indicators
- The ability to view near-term historical data

One or more enhanced 3270 interface address spaces can be deployed on any given sysplex. A monitoring agent or a remote monitoring server must run in the same sysplex as the interface to enable discovery and communications with the associated hub monitoring server.

The OMEGAMON enhanced 3270 user interface requires that at least one hub monitoring server be configured. The scope of the data displayed by the interface is determined by the scope of the associated hub monitoring server. If a hub monitoring server configuration connects agents from systems that span multiple sysplexes, the enhanced 3270 interface displays data from multiple sysplexes.

IBM Z OMEGAMON for JVM provides a set of workspaces and associated help panels for the OMEGAMON (enhanced 3270UI). The enhanced 3270UI workspaces use the same attribute groups that are detailed in the following topics.

The Tivoli Enterprise Portal

The Tivoli Enterprise Portal can be accessed by using the Java Web Start client, a supported browser client, or a Java™-based desktop client. Using the Tivoli Enterprise Portal interface requires defining and running the Tivoli Enterprise Portal Server, a hub Tivoli Enterprise Monitoring Server, and potentially one or more remote monitoring servers.

The Tivoli Enterprise Portal provides the following features:

- The ability to merge the data from multiple agents or multiple systems into a single workspace for business views and unit of work views
- The ability to define specific conditions that should be monitored (situations)

- The ability to define colored highlights and sounds to alert operators when events occur
- Proactive automation to respond to situations
- Highly customizable graph and table views
- Integral historical data collection, reporting, and warehousing
- Customizable historical reporting
- Integration with other Tivoli® and IBM® products, including single sign-on
- Event forwarding to event management systems LDAP authentication

The Java Web Start client offers the same graphical user interface as the desktop client with a footprint that is equally as small as the browser client. The Java Web Start client does not have the overhead and loss of real estate as the browser client. In addition, the Java Web Start and browser clients have the added benefits that no installation is required on the user client platform and that time to value (TTV) and total cost of ownership (TCO) are much improved.

IBM Z OMEGAMON for JVM provides a set of workspaces and situations for the Tivoli® Enterprise Portal. You can customize the workspaces and situations and you can create your own. IBM Z OMEGAMON for JVM also provides attribute groups that define the JVM properties that are available to view in your workspaces and that you use to drive situations. The predefined workspaces, attribute groups, and situations that IBM Z OMEGAMON for JVM provides are described in the following topics. The material is also available as online help in the Tivoli Enterprise Portal when you install application support for IBM Z OMEGAMON for JVM.

User interface help reference

You can use the Tivoli Enterprise Portal and the OMEGAMON enhanced 3270 user interface to view monitored Java virtual machine (JVM) data.

The Tivoli Enterprise Portal is a graphical user interface that presents JVM data in a set of predefined workspaces that are provided by IBM Z OMEGAMON for JVM. With the Tivoli Enterprise Portal, you view tables of data and you can also view graphical representations of data. You can link from one workspace to another to do in-depth analysis of the data. The workspaces that IBM Z OMEGAMON for JVM provides for the Tivoli Enterprise Portal are described in [“Workspaces” on page 3](#).

An alternative interface is the OMEGAMON enhanced 3270 user interface (enhanced 3270UI), which can be used with any 3270 emulator and is designed for quick problem solving. IBM Z OMEGAMON for JVM provides enhanced 3270UI workspaces that are comparable to the ones for the Tivoli Enterprise Portal, but data displays in subpanels. You use the **Options Menu** to quickly drill for additional information and you can also use the Zoom feature for context-sensitive navigation. IBM Z OMEGAMON for JVM also provides help for its enhanced 3270UI workspaces. The help explains the workspace in detail and in many cases, provides advice and tips that help you analyze the data and solve problems with your JVMs.

When you open an IBM Z OMEGAMON for JVM workspace, chart and table views in a Tivoli Enterprise Portal workspace and subpanels in an enhanced 3270UI workspace use *queries* that specify the data to be displayed. The queries are routed through the Tivoli Management Services infrastructure to the appropriate IBM Z OMEGAMON for JVM monitoring agent. The monitoring agent collects the JVM data, which is returned through the infrastructure and displayed in the Tivoli Enterprise Portal or enhanced 3270UI client that requested it.

The characteristics or properties of the logical and physical objects that are monitored by IBM Z OMEGAMON for JVM are known as *attributes*. A set of related attributes is known as an *attribute group*. Attributes are used to define the queries that specify the data to be displayed in the workspaces. Attributes are also used to describe *situations*, or conditions, that can trigger events.

With the proper user authority, you can tailor Tivoli Enterprise Portal views to display critical or warning indicators when monitored values reach specified thresholds, and to filter incoming data so that you see only information that you are interested in. You can add additional views to existing workspaces or create your own workspaces. You can also define your own queries and your own situations by using IBM Z OMEGAMON for JVM attributes.

For more information about the enhanced 3270UI, see *OMEGAMON enhanced 3270 user interface* in *OMEGAMON XE shared documentation Version 6.3.0 Fix Pack 2 and above*. For information about the Tivoli Enterprise Portal, including how to manage and regulate access to workspaces, how to work with queries, situations, and take action commands, how to collect historical data, and more, see *IBM Tivoli Monitoring: Tivoli Enterprise Portal User's Guide* and *IBM Tivoli Monitoring: Administrator's Guide*. In addition, many of the products in the OMEGAMON XE family contain general information about using the Tivoli Enterprise Portal that might help you understand how to work with IBM Z OMEGAMON for JVM resources in the Tivoli Enterprise Portal interface.

Workspaces

IBM Z OMEGAMON for JVM provides predefined workspaces that you can use to view information in the Tivoli Enterprise Portal about the Java virtual machines that you are monitoring. The workspaces contain table views and graphs that you can use to obtain information about the monitored JVMs and about various aspects of your JVM environments.

The workspaces deliver detailed, current data that enables you to analyze your monitored JVMs. You can use the table views and graphs within each workspace to complete the following tasks:

- Investigate attribute information that relates to a change in state
- Monitor the performance of your JVMs and evaluate performance issues
- Verify operational characteristics of your JVM environments
- Determine the most effective threshold values for situations you create

You access the workspaces from the Navigator Physical view in the Tivoli Enterprise Portal. Certain workspaces also provide links to subsidiary (secondary) workspaces. A set of historical workspaces are also provided which deliver JVM CPU, garbage collection, and health data for recent historical intervals.

Address Space Memory workspace

The **Address Space Memory** workspace the utilization of the three areas of a z/OS address space virtual memory:

- 24-bit virtual storage (below the 16MB line)
- 31-bit virtual storage (above the 16MB line but below the 2GB bar)
- 64-bit virtual storage (above the 2GB bar)

If free storage in any of these areas is exhausted, it will result in an OutOfMemory condition and the JVM will terminate.

Use this workspace to identify whether virtual storage in any of these areas is approaching exhaustion. As a general guideline, if free storage in any of these areas falls below 10% of the total available, an OutOfMemory condition may occur. Use the OMEGAMON history report to determine whether this area of the region has been growing which may indicate a memory leak.

This workspace also displays **Language Environment Heap data**. Language Environment Heap statistics are disabled by default, and show zeroes in the Native Memory workspace.

To enable Language Environment Heap statistics, specify the following system property in the JVM options of the monitored JVM:

```
-Dkjj.leheapstats.enabled=y
```

Completed Job Summary workspace

Use the **Completed Job Summary** workspace to review monitored JVMs that have completed recently.

The source of this data is the JVM Health history stored in the near-term history (NTH) persistent data store. You must enable history for the JVM Health table either in the enhanced 3270UI View pull down menu, History menu item, or in the Tivoli Enterprise Portal (TEP) user interface in order to collect this data.

The summary displays the worst case samples of key performance metrics collected in JVM Health history. Select the link on a result row to view all the samples collected during the execution of that job.

Note that the NTH data store has a finite size. Eventually the oldest samples are removed to make room for newer records. The duration that completed job samples are retained depends on several factors:

- The size and number of persistent data store data sets.
- The number of jobs being monitored.
- The frequency of history sample collections.
- The number of other tables being collected for history.

Essentially, the more data samples that are taken across all tables, the shorter the period the individual job history samples are retained.

Completed Job Detail workspace

The **Completed Job Summary** displays all the health history for a single completed job.

Each row is a snapshot of the JVM Health at the time. Averages and counts are calculated based on the history collection interval.

CPU Statistics workspace

The **CPU Statistics** workspace displays processor information for each monitored Java virtual machine (JVM).

For each monitored JVM, processor time and usage on general-purpose processors and Integrated Facility for Applications (IFA) specialized processors (zIIPs and zAAPs) is displayed. Processing that was IFA-eligible, but instead ran on general-purpose processors is also shown.

The **CPU Statistics** workspace uses the JVM CPU attribute group to populate the following views:

- CPU Times

This bar chart shows processor times for each JVM.

- CPU Percentages

This bar chart shows processor usage for each JVM.

- CPU Statistics

This table view lists the CPU metrics for each monitored JVM. From this view, you can navigate to the **Environment Data**, **Lock Statistics**, **Thread Statistics**, and **Historical CPU Statistics** workspaces.

Environment Data workspace

The **Environment Data** workspace displays information about the Java environment in which a selected Java virtual machine (JVM) is running.

Overview and detailed environment data is displayed in the workspace. You can use the information in the workspace to review the configuration of the JVM environment and identify incorrect or inconsistent deployments.

The **Environment Data** workspace uses the JVM Environment Data and JVM Environment Data Details attribute groups to populate the following views:

- Environment Data

This table displays general information about the Java environment for the JVM. The single row identifies the JVM and shows the version of Java that it uses and the UNIX System Services path for the Java installation. It also shows the type of z/OS subsystem that is hosting the JVM.

You can use the information in the table to verify that the JVM is using the correct version of Java.

- System Properties

This table displays a list of each system property that is defined in the JVM environment. The first column in the view shows the name of a property. The second column displays the value. You can use the information to verify that a specific system property is set to the right value.

- Environment Variables

This table displays a list of each environment variable that is defined in the JVM environment. The first column in the view shows the name of a variable. The second column displays the value. You can use the information to verify that a specific environment variable is set to the right value to ensure proper operation of the JVM.

- JVM Parameters

This table displays a list of the parameters that are defined in the JVM environment. Parameters affect how the JVM operates. They can be specified on the command line when you start the JVM or by using environment variables.

Each parameter is displayed on a separate row in the view. The first column shows the position of the parameter that is displayed in the second column. You can use the information to verify that the parameters for the JVM are set properly.

- Command Line

This table displays a list of the items that were on the command line when the JVM started. Each command line argument is displayed on a separate row. The first column in the view shows the position in the command line of the argument that is displayed in the second column.

Command line arguments can affect the behavior of the JVM and the applications that run in it. Use the information in this table to verify that the right arguments are set.

- Classpath

This table provides a list of the directories, JAR files, and .zip files that are on the classpath that is used by the JVM. Each classpath entry is displayed on a separate row. The first column in the view shows the position in classpath of the entry that is displayed in the second column.

You can use the information to verify that the correct entries are present and that they are specified in the right order. You can also determine whether erroneous entries are listed, which might cause applications that are running in the JVM to operate incorrectly.

- Boot Classpath

This table provides a list of the items that are on the boot classpath that is used by the JVM. Each boot classpath entry is displayed on a separate row. The first column in the view shows the position in the boot classpath of the entry that is displayed in the second column.

You can use the information to verify that the correct entries are present. You can also determine whether erroneous entries are listed, which might cause applications that are running in the JVM to operate incorrectly. A boot classpath that is not correctly set might indicate that Java was not installed properly or that a command line override caused the JVM to be mis-configured.

Garbage Collection Details workspace

The **Garbage Collection Details** workspace displays garbage collection detail information for a selected Java virtual machine (JVM).

For a monitored JVM, garbage collection (GC) rates and other metrics such as GC pause times are displayed. The metrics are calculated over the last 5 minutes.

The **Garbage Collection Details** workspace uses the JVM Garbage Collection attribute group to populate the following views:

- Garbage Collection Rate

This chart shows the rate of garbage collections per minute.

- Pause Times

This bar chart shows various garbage collection pause time metrics. You can use the chart to compare, for example, the average garbage collection pause time to the maximum garbage collection pause time or the average nursery garbage collection pause time.

- Collection Intervals

This bar chart shows various garbage collection interval metrics. You can use the chart to compare, for example, the overall average garbage collection interval to the average global garbage collection interval or the average nursery garbage collection interval.

- Occupancy after Collection

This chart shows the average percentage of heap that was in use after garbage collection.

- Heap Usage

This bar chart shows various metrics about heap usage. You can use the chart to compare, for example, the maximum heap size to the maximum used heap size.

- Garbage Collection Counts

This bar chart shows counts for various garbage collection metrics. You can use the chart to compare, for example, the total number of garbage collections that occurred to the number of garbage collections that were forced by the system or the number of global garbage collections.

- Garbage Collection Details

This table view displays a single row with the garbage collection statistics for the selected JVM. From this view, you can navigate to the **Environment Data**, **Lock Statistics**, **Thread Statistics**, and **Historical Garbage Collection Statistics** workspaces.

Garbage Collection Statistics workspace

The **Garbage Collection Statistics** workspace displays garbage collection information for each monitored Java virtual machine (JVM).

For each monitored JVM, garbage collection (GC) rates and other metrics such as GC pause times are displayed. The metrics are calculated over the last 5 minutes.

The **Garbage Collection Statistics** workspace uses the JVM Garbage Collection attribute group to populate the following views:

- Garbage Collection Rates

This bar chart shows the rate of garbage collections per minute for each JVM.

- Proportion of Time in Garbage Collection Pauses

This bar chart shows the percentage of time that each JVM spent in garbage collection pauses.

- Occupancy after Garbage Collection

This bar chart shows for each JVM the average percentage of heap that was in use after garbage collection.

- System Garbage Collections

This bar chart shows the number of garbage collections for each JVM that were forced by the system.

- Garbage Collection Statistics

This table view lists GC metrics for each monitored JVM. From this view, you can navigate to the **Garbage Collection Details** workspace to see the GC details for a JVM. You can also navigate to the **Environment Data**, **Lock Statistics**, **Thread Statistics**, and, if available, **Historical Garbage Collection Statistics** workspaces.

Note: The garbage collection mode for a JVM cannot be determined until garbage collection occurs.

Health Summary workspace

The **Health Summary** workspace displays key performance metrics for each monitored Java virtual machine (JVM).

For each monitored JVM, garbage collection (GC) rates and other key performance metrics are displayed. The metrics are calculated over the last 5 minutes.

The **Health Summary** workspace uses the JVM Health Summary attribute group to populate the following views:

- Garbage Collection Rates

This bar chart shows the rate of garbage collections per minute for each JVM.

- Proportion of Time in GC Pauses

This bar chart shows the percentage of time that each JVM spent in garbage collection pauses.

- Occupancy after Collection

This bar chart shows the average percentage of heap that was in use after garbage collection for each JVM.

- System Garbage Collections

This bar chart shows the number of garbage collections for each JVM that were forced by the system.

- General CPU Usage

This bar chart shows the percentage of the processing time for each JVM that occurred on general-purpose processors.

- zIIP on CP Usage

This bar chart shows the percentage of the processing time for each JVM that occurred on general-purpose processors, but was eligible to run on specialized zIIP processors.

- Thread Counts

This bar chart shows the number of active threads that are in each JVM.

- Threads Blocked

This bar chart shows the number of threads that are blocked in each JVM.

- Lock Utilization

This bar chart shows the highest lock usage value for each JVM. The lock usage value is the percentage of time that a lock was held.

- Lock Missed

This bar chart shows the highest lock missed percentage for each JVM. The lock missed percentage reflects the number of waits for a lock.

- Average Lock Hold Times

This bar chart shows the highest average lock hold time for each JVM. The average lock hold time is the average amount of time that a lock was held.

- Health Summary

This table view shows key performance metrics for each monitored JVM. From this view, you can navigate to the **Environment Data**, **Lock Statistics**, **Thread Statistics**, and **Historical Health Summary** workspaces.

JVM Overview workspace

The **JVM Overview** workspace displays a list of the known Java virtual machines (JVMs) that are on the LPAR on which the OMEGAMON Monitoring for JVM collector is running.

This workspace lists the JVMs that are being monitored by the OMEGAMON Monitoring for JVM collector and the JVMs on the LPAR that are not being monitored by the collector. In addition, the worst statistics among the JVMs that are being monitored by the collector are displayed. You can quickly see the worst performance values before you navigate to other OMEGAMON Monitoring for JVM workspaces that provide more details that you can use for problem solving. By default, the values are calculated over the last 5 minutes.

The **JVM Overview** workspace uses the JVM Auto Discovery and JVM Overview attribute groups to populate the following views:

- JVMs Monitored by This Collector

This table view shows a list of the JVMs that are configured for monitoring by OMEGAMON Monitoring for JVM and are providing data to the collector. You can access comprehensive performance data for these JVMs from the other OMEGAMON Monitoring for JVM workspaces that are in the Navigator tree.

- **JVMs Not Monitored by This Collector**

This table view shows a list of the JVMs that are on the LPAR, but are not configured for monitoring by OMEGAMON Monitoring for JVM. Enhanced metrics cannot be provided for these JVMs until they are configured for monitoring. The table displays information that identifies the JVMs, which you can use to determine which JVMs must be configured for monitoring.

Note: If you run more than one OMEGAMON Monitoring for JVM collector on an LPAR, JVMs that are reporting to a different collector appear in this list. To see the metrics for those JVMs, select the node in the Navigator for that collector.

- **JVM Overview**

This table view shows a summary of the worst key performance metrics among the JVMs being monitored by the collector. Metrics such as the highest garbage collection rate among the JVMs and highest processor usage among the JVMs are displayed. From this view, you can navigate to workspaces that provide details about the JVM that reported a specific metric.

Lock Statistics workspace

The **Lock Statistics** workspace displays Java application lock information for a selected Java virtual machine (JVM).

Statistics about the locks with the highest usage for the JVM, up to a maximum of 50, are displayed in the workspace. You can use this workspace to analyze lock contention in a JVM.

The **Lock Statistics** workspace uses the JVM Locks attribute group to populate the following views:

- **Lock Utilization**

This chart shows the lock with the highest usage. It was held for the highest percentage of time.

- **Lock Missed**

This chart shows the lock with the highest percentage of waits.

- **Average Lock Hold Time**

This chart shows the lock that was held for the highest average amount of time.

- **Lock Statistics**

This table view lists Java application lock information for the selected JVM. The locks with the highest usage, up to a maximum of 50, are listed.

Native Memory workspace

The **Native Memory** workspace displays the native memory that different areas or categories in the JVM use.

The **Native Memory** workspace uses the JVM Native Memory attribute group to populate the following views:

- **Native Memory Statistics**

This table view provides the memory for the application process by the operating system. The memory is used for heap storage and other purposes. The information that is shown in this table view represents different areas or categories in the JVM that are using the native memory.

The Language Environment Heap statistics are disabled by default, and show zeroes in the Native Memory workspace. To enable Language Environment Heap statistics, specify system property - `Dkjj.1eheapstats.enabled=y` in the JVM options of the monitored JVM.

Thread Statistics workspace

The **Thread Statistics** workspace displays information about the threads that are running on a selected Java virtual machine (JVM).

The **Thread Statistics** workspace uses the JVM Threads attribute group to populate the following views:

- CPU Percentage

This chart shows the 5 most CPU-consuming threads.

- Threads Blocked

This chart shows the number of threads in the JVM that are blocked.

- Thread Statistics

This table view lists information about the threads that are running in the selected JVM. If a thread is blocked, the name of the contending object and the name of the thread that owns the contending object are displayed. The stack trace information can be used to determine the source code that is running.

z/OS Connect Slowest Requests workspace

The **z/OS Connect Slowest Requests** workspace displays the five worst performing requests over the last 5 minutes for a specific z/OS Connect service.

Because the **z/OS Connect Slowest Requests** workspace displays only the five worst performing requests over the last 5 minutes, the following rules apply:

- If there were no requests over the last 5 minutes, the **z/OS Connect Slowest Requests** workspace displays nothing.
- If there were fewer than five requests over the last 5 minutes, the **z/OS Connect Slowest Requests** workspace displays all these requests.

The **z/OS Connect Slowest Requests** workspace uses the JVM Request Details attribute group to populate the following views:

- Response Time

This graphical view provides the Response Time for the 5 worst performing requests over the last 5 minutes for a specific z/OS Connect service. This view can be used to identify requests that might have resulted in an error, or to determine whether a service is underperforming or being overworked.

- z/OS Connect Slowest Requests Statistics

This table view provides performance metrics for the 5 worst performing requests over the last 5 minutes for a specific z/OS Connect service.

To view historical data for the selected service, select a time span by using the **Set Historical Navigation Mode** icon on the right side of the main toolbar at the top of the Tivoli Enterprise Portal.

If you set the time span before you navigate to the **z/OS Connect Slowest Requests** workspace, you don't have to set the time span again.

If the time span that you select covers more than one historical collection interval, you can see highlighted navigation arrows on the scroll bar on the right side of the **z/OS Connect Slowest Requests** view, with which you can navigate to request data from earlier or later historical collection intervals.

z/OS Connect Request Details workspace

The **z/OS Connect Request Details** workspace displays details for a specific z/OS Connect Service, and provides aggregate data for the service on requests that have been executed in the last 5 minutes. You can use this workspace to navigate to the **z/OS Connect Slowest Requests** workspace to display the 5 worst performing requests over the last 5 minutes for this service.

The **z/OS Connect Request Details** workspace uses the JVM Request Summary attribute group to populate the following views:

- Average Response Time

This graphical view provides the Average Response Time, in seconds, for each service that has been executed in the last 5 minutes. Services with a high Average Response Time might indicate that z/OS Connect servers or subsystems that are performing poorly.

- Maximum Response Time

This graphical view provides the Maximum Response Time, in seconds, for each service that has been executed in the last 5 minutes. Services with a high Maximum Response Time might indicate requests that have resulted in errors or poorly performing z/OS Connect servers or subsystems.

- Minimum Response Time

This graphical view provides the Minimum Response Time, in seconds, for each service that has been executed in the last 5 minutes.

- z/OS Connect Request Details

This table view provides performance metrics for the selected service. From this view you can navigate to the **z/OS Connect Slowest Requests** workspace to display the 5 worst performing requests over the last 5 minutes for this service.

To view historical data for the selected service and retain the historical context when you navigate to the **z/OS Connect Slowest Requests** workspace, select a time span by using the **Set Historical Navigation Mode** icon on the right side of the main toolbar at the top of the Tivoli Enterprise Portal.

z/OS Connect Request Summary workspace

The **z/OS Connect Request Summary** workspace displays all of the services that have been called within the lifetime of the z/OS Connect server, and provides aggregate data for each service on requests that have been executed in the last 5 minutes. You can use this workspace to identify services that perform slowly or are overworked.

The **z/OS Connect Request Summary** workspace uses the JVM Request Summary attribute group to populate the following views:

- Average Response Time

This graphical view provides the Average Response Time, in seconds, for each service that has been executed in the last 5 minutes. Services with a high Average Response Time might indicate that z/OS Connect servers or subsystems that are performing poorly.

- Maximum Response Time

This graphical view provides the Maximum Response Time, in seconds, for each service that has been executed in the last 5 minutes. Services with a high Maximum Response Time might indicate requests that have resulted in errors or poorly performing z/OS Connect servers or subsystems.

- Request Count

This graphical view provides the number of requests for each service that has been executed in the last 5 minutes. Because all z/OS Connect requests expire after 5 minutes, this value might vary over time. Services with high Request Count values might indicate overworked z/OS Connect servers.

- z/OS Connect Request Summary Statistics

This table view provides performance metrics for each service that is defined in the z/OS Connect server that has been executed within the lifetime of the z/OS Connect server. This view can be used to identify services that perform slowly or are overworked.

Extended z/OS Connect Request Details workspaces

The detail workspaces contain the individual requests that comprise the selected summary row.

Note: z/OS Connect metrics are stored in a z/OS System log stream. Your system programmer may define the log stream offload datasets to be kept only for a finite amount of time. You cannot generate summary reports for request data beyond this retention period.

The following data are shown:

- Event Time

The date and time that request entered z/OS Connect. The date is in the following format: mm-dd-yyyy.

- API Name

The name of the API associated with the requests or one of the following reserved names:

ADMIN A z/OS Connect administration request

*SERVIC
E* A service invoked directly

- Method

The name of the HTTP method with which this request was made, for example, GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS.

- Service Name

The name of the service invoked by the request. For administration API queries, this value may be N/A. For requests that do not invoke a service, either because there was an error or the service is not active, the value *NONE* is used.

- Total Req Time

The total amount of time between the request being received by z/OS Connect and a response being returned to the request client.

- Z/OS Conn Time

The amount of time the request spent in z/OS Connect in seconds.

- SoR Resp Time

The amount of time the request spent being processed by the System of Record.

- HTTP Code

The HTTP status code for the response. If the asynchronous request processing has timed out, the value returned for HTTP status code indicates how the asynchronous work was processed by z/OS Connect EE but does not reflect the response that the HTTP client received.

- Timeout

Indicates whether this request was subject to asyncTimeout. The HTTP Code may be 200 as the work completed successfully in the System of Record, but if the asynchronous request timeout expired before the request had finished processing, the HTTP client will have actually received the HTTP status code 503 (Gateway Timeout).

- SoR ID

SoR ID is a string representing the System of Record application. For the CICS service provider, this is the fully qualified CICS applied. For the IMS service provider, it is the datasource name. For the REST client service provider, it is the IP address/hostname and port of the target API endpoint.

- SoR Ref

The configured connection reference that identifies the connection to the target System of Record. This is either the value set in the associated service archive file or the value specified in a policy override that applies to this request. For the CICS service provider, it is the ID attribute of the `zosconnect_cicsIpicConnection` element. For the IMS service provider, it is the ID attribute of the `imsmobile_imsConnection` element. For the REST client service provider, it is the ID attribute of the `zosconnect_zosConnectServiceRestClientConnection` element.

- SoR Resource

A string that contains a specific service provider information, which identifies the resource that is invoked on the System of Record. For the CICS service provider, it is in the following format: <transid>, <program>. For the IMS service provider, it is in the transaction code. For the REST client service provider, it is the URI of the target system resource.

- Request URI

The part of this request's URL from the protocol name up to the query string in the first line of the HTTP request.

- Port

The URL port used for this request interval.

- User ID

Name of the authenticated user making the request. May be blank for non-authenticated requests.

- Remote Address

The Internet Protocol (IP) address of the client or last proxy that sent the request.

- Request Length

The length of the JSON request payload.

- Response Length

The length of the JSON response payload.

- Operation

The operation in the API associated with this request. May be '*NONE*' for administration requests, or requests that invoke a service directly.

- Provider Name

The name of the service provider which processed this request. May show as N/A for administration requests or '*NONE*' for requests that did not successfully invoke a service.

Extended z/OS Connect Request Summary workspaces

These workspaces summarize z/OS Connect request metrics for the selected server.

Note: z/OS Connect metrics are stored in a z/OS System log stream. Your system programmer may define the log stream offload datasets to be kept only for a finite amount of time. You cannot generate summary reports for request data beyond this retention period.

The following attributes are shown:

- HTTP Method

The HTTP method used to make the API request. One of: GET, PUT, POST, PATCH, DELETE, HEAD, OPTIONS.

- Request Count

The total number of requests that completed.

- Error Count

The number of requests that ended in error. Any HTTP error code of 300 or higher is considered an error.

- Timeout Count

The number of requests ended by `asyncRequestTimeout`.

- Resp Time Min

The minimum total response time observed during the interval.

- Resp Time Max

The maximum total response time observed during the interval.

- Resp Time Avg

The average total response time for all requests of this summary type.

- Resp Time StdDev

The standard deviation of total response time during the summary interval.

- zOS Connect Time Min

The minimum time observed for a request during the summary interval.

- zOS Connect Time Max

The maximum time observed for a request during the summary interval.

- zOS Connect Time Avg

The average time that requests spent in z/OS Connect.

- zOS Connect Time StdDev

The standard deviation of z/OS Connect time during the summary interval.

- SoR Time Min

The minimum amount of time spent by a request making a call to the System of Record.

- SoR Time Max

The maximum amount of time a request spent making a call to the System of Record (SoR).

- SoR Time Avg

The average time requests spent making calls to the System of Record (SoR).

- SoR Time StdDev

The standard deviation of SoR time during the summary.

- API Name

The name of the API used or one of the following reserved names:

ADMIN A z/OS Connect administration request

*SERVIC A service invoked directly

E*

- Service Name

The name of the service used to invoke requests

- SoR ID

SoR ID is a string representing the System of Record application. For the CICS service provider, this is the fully qualified CICS applied. For the IMS service provider, it is the datasource name. For the REST client service provider, it is the IP address/hostname and port of the target API endpoint.

- SoR Reference

The configured connection reference that identifies the connection to the target System of Record. This is either the value set in the associated service archive file or the value specified in a policy override that applies to this request. For the CICS service provider, it is the ID attribute of the `zosconnect_cicsIpicConnection` element. For the IMS service provider, it is the ID attribute of the `imsmobile_imsConnection` element. For the REST client service provider, it is the ID attribute of the `zosconnect_zosConnectServiceRestClientConnection` element.

- SoR Resource

A string that contains a specific service provider information, which identifies the resource that is invoked on the System of Record. For the CICS service provider, it is in the following format: `<transid>`, `<program>`. For the IMS service provider, it is in the transaction code. For the REST client service provider, it is the URI of the target system resource.

Attributes

Attributes are characteristics or properties of Java virtual machine (JVM) resources that are monitored by IBM Z OMEGAMON for JVM. Related attributes are organized into *attribute groups* (which are referred to as *tables* in the context of historical data collection).

Attributes are used to define the queries that collect the JVM information that is displayed in tables and charts in OMEGAMON Monitoring for JVM Tivoli Enterprise Portal workspaces and in OMEGAMON

enhanced 3270 user interface workspaces. You also use attributes to create situations that trigger alerts and automate actions in response to specified conditions.

Various predefined Tivoli Enterprise Portal workspaces are included in IBM Z OMEGAMON for JVM. In these workspaces, a table view corresponds to a group of attributes and the columns in the table correspond to attributes.

The following table shows the relationships between the predefined workspaces and the attribute groups. The workspaces are listed in alphabetical order.

<i>Table 1. OMEGAMON Monitoring for JVM Tivoli Enterprise Portal workspaces and related attribute groups</i>	
Workspace	Related Attribute Group
CPU Statistics	“JVM CPU attributes” on page 16
Environment Data	“JVM Environment Data attributes” on page 16 “JVM Environment Data Details attributes” on page 17
Garbage Collection Details	“JVM Garbage Collection attributes” on page 17
Garbage Collection Statistics	“JVM Garbage Collection attributes” on page 17
Health Summary	“JVM Health Summary attributes” on page 18
Historical CPU Statistics	“JVM CPU attributes” on page 16
Historical Garbage Collection Statistics	“JVM Garbage Collection attributes” on page 17
Historical Health Summary	“JVM Health Summary attributes” on page 18
JVM Overview	“JVM Auto Discovery attributes” on page 14 “JVM Overview attributes” on page 20
Lock Statistics	“JVM Locks attributes” on page 19
Native Memory	“JVM Native Memory attributes” on page 20
Thread Statistics	“JVM Threads attributes” on page 21
z/OS Connect Request Details	“JVM Legacy z/OS Connect Request Summary attributes” on page 23
z/OS Connect Request Summary	“JVM Legacy z/OS Connect Request Summary attributes” on page 23
z/OS Connect Slowest Requests	“JVM Legacy z/OS Connect Request Details attributes” on page 22

JVM Auto Discovery attributes

ASID

Collector ID

Full Version

Java Home

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Monitored Valid values include Yes and No.

Process ID

SMF ID

Subsystem Type Valid values include CICS, CICSTG, DB2, IMS, Liberty, Standalone, WebSphere, z/OSMF, ODM, UCD, Unknown, and N/A. N/A means that the Java Agent has not yet determined the subsystem type because the Java Agent is not completely initialized.

JVM Completed Job Summary attributes

Managed System

SMF ID

Collector ID

Job Name

ASID

Process ID

Subsystem Type

Max Garbage Collection Rate

Max Garbage Collection Pause

Max Heap Occupancy

Max Lock Miss Percent

Max Lock Util

Max Lock Hold Time

Max General CPU

Max zIIP on CP

Max Thread Count

Max Threads Blocked

Application

Sample Count

Start Time

JVM Completed Job Detail attributes

Managed System

SMF ID

Collector ID

Write Time

Job Name

ASID

Process ID

Subsystem Type

GCs per Minute

Percent Time in GC Pauses

Heap Occupancy
System GC Count
Lock Missed Percent
Lock Util Percent
Avg Lock Hold Time
General CPU Percent
zIIP on CP Percent
Thread Count
Threads Blocked
Application
JVM Profile
Application Version
Start Time
64-Bit Memory Free Percent
64-Bit Memory Free
24-Bit Memory Free
24-Bit Memory Free Percent
31-Bit Memory Free
31-Bit Memory Free Percent
JVM CPU attributes

ASID
Collector ID
Elapsed Time
General CPU Percent
General CPU Time
zIIP Percent
zIIP Time
zIIP on CP Percent
zIIP on CP Time
Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Process ID
SMF ID

JVM Environment Data attributes

ASID

Collector ID

Full Version

Java Home

Job Name

Process ID

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Processors

SMF ID

Subsystem Type Valid values include CICS, CICSTG, DB2, IMS, Liberty, Standalone, WebSphere, z/OSMF, ODM, UCD, and Unknown.

Version

JVM Version

JVM Environment Data Details attributes

ASID

Collector ID

Environment Data Type Valid values include Classpath, Boot Classpath, System Properties, Command Line, Environment Variables, and Java Parameters.

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Name If the type of environment data is not a KEY=VALUE pair, then Name might be blank.

Order Number The first property in the series is numbered 1. The second property is numbered 2, and so on.

Process ID

SMF ID

Value

JVM Garbage Collection attributes

Allocation Failure Count

ASID

Avg Collection Interval

Avg Global Collection Interval

Avg Global Pause Time

Avg Nursery Collection Interval

Avg Nursery Pause Time

Avg Occupancy Percent

Avg Partial Collection Interval This attribute applies to a balanced garbage collection policy.

Avg Partial Pause Time This attribute applies to a balanced garbage collection policy.

Avg Pause Time**Collection Count**

Collection Mode Valid values include BALANCED, GENCON, OPTAVGPAUSE, OPTTHRUPUT, and UNKNOWN. The BALANCED mode is available only on 64-bit JVMs.

Collections per Minute**Collector ID****Concurrent Collection Count****Final References Cleared Count****Global Collection Count**

Global Mark Phase Avg Collection Interval This attribute applies to a balanced garbage collection policy.

Global Mark Phase Avg Pause Time This attribute applies to a balanced garbage collection policy.

Global Mark Phase Avg Collections per Cycle This attribute applies to a balanced garbage collection policy.

Global Mark Phase Collection Count This attribute applies to a balanced garbage collection policy.

Global Mark Phase Cycle Count This attribute applies to a balanced garbage collection policy.

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Max Heap Size**Max Pause Time****Max Used Heap****Mean Heap Size****Mean Used Heap****Min Heap Size****Min Pause Time****Min Used Heap****Nursery Amount Flipped****Nursery Collection Count**

Partial Collection Count This attribute applies to a balanced garbage collection policy.

Percent Time in Collection Pauses**Percent Time Unpaused****Process ID****Sample Period****SMF ID**

Synchronous Collection Count This attribute applies to a balanced garbage collection policy.

System Collection Count**JVM Health Summary attributes****Application**

Application Version
ASID
Avg Lock Hold Time
Heap Occupancy
Collector ID
GCs per Minute
General CPU Percent
zIIP on CP Percent
Job Name
JVM Profile
Lock Missed Percent
Lock Util Percent
Managed System
24-Bit Free Percent
24-Bit Free
31-Bit Free Percent
31-Bit Free
64-Bit Free Percent
64-Bit Free
Percent Time in GC Pauses
Process ID
SMF ID
Start Time
Subsystem Type
System GC Count
Thread Count
Threads Blocked
JVM Locks attributes

ASID

Avg Hold Time

Collector ID

Get Count

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Missed Percent

Monitor Name

Process ID

Recursive Acquires

Slow Gets

SMF ID

Utilization Percent

JVM Native Memory attributes

Allocated Deep

Allocated Shallow

Bytes Deep

Bytes Shallow

ASID

Category Name

Category Parent

Collector ID

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Process ID

SMF ID

Tree Level The level is used to identify the distance from the highest parent node.

JVM Overview attributes

The key performance indicators highlight the worst performing Java virtual machine (JVM) for each category, among the JVMs monitored by the collector on the LPAR.

Collector ID

Highest Avg Lock Hold Time

Highest Avg Lock Hold Time PID

Heap Occupancy

Highest Avg Occupancy Percent PID

GC Rate per Minute

Highest GCs per Minute PID

General CPU Percent

Highest General CPU Percent PID

zIIP on CP Percent

Highest zIIP on CP Percent PID

Highest Lock Missed Percent

Highest Lock Missed Percent PID

Lock Util Percent

Highest Lock Utilization Percent PID

Highest Percent Time in Garbage Collection Pauses

Highest Percent Time in Garbage Collection Pauses PID

System GC Count

Highest System GC Count PID

Thread Count

Highest Thread Count PID

Highest Threads Blocked

Highest Threads Blocked PID

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Monitored JVM Count

Not Monitored JVM Count

SMF ID

SMF ID Collector ID

JVM Threads attributes

ASID

Collector ID

Contending Object

Contending Thread ID

Contending Thread Name

CPU Time

CPU Percent

Job Name

Locks Owned

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Number Monitors Owned

Process ID

SMF ID

Stack Trace

State Valid values include the following settings.

BLOCKED

The thread is blocked waiting for a monitor lock.

NEW

The thread is not started.

RUNNABLE

The thread is running in the Java virtual machine.

TERMINATED

The thread exited.

TIMED WAITING

The thread is waiting for a specified amount of time for another thread to perform an action.

WAITING

The thread is waiting indefinitely for another thread to perform an action.

UNKNOWN

The thread state is unknown.

Thread ID**Thread Name****JVM Virtual Memory attributes****Managed System****SMF ID****Collector ID****Job Name****ASID****Process ID****Heap Size****Heap Allocated****Heap Free****24-Bit Memory Size****24-Bit Memory User****24-Bit Memory Authorized****24-Bit Memory Free****24-Bit Memory Used Percent****24-Bit Memory Free Percent****31-Bit Memory Size****31-Bit Memory User****31-Bit Memory Authorized****31-Bit Memory Free****31-Bit Memory Used Percent****31-Bit Memory Free Percent****64-Bit Memory Limit****64-Bit Memory Used****64-Bit Memory Free****64-Bit Memory Used Percent****64-Bit Memory Free Percent****JVM Legacy z/OS Connect Request Details attributes****ASID****Collector ID****Event Time**

Host

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Method Valid methods include GET, PUT, POST, DELETE, PATCH, and HEAD.

Port

Process ID

Query String

Remote Address

Request ID

Request Length

Response Time

Response Length

Service Name

Service ID The service ID isn't displayed in the Tivoli Enterprise Portal or the OMEGAMON enhanced 3270 user interface.

SMF ID

JVM Legacy z/OS Connect Request Summary attributes

ASID

Avg Response Time

Avg Response Length

Avg Request Length

Collector ID

Host

Job Name

Managed System The value is a text string of up to 32 alphanumeric characters that combines the Collector ID and the SMF ID in the following format: Collector_ID:SMF_ID:JVM.

Method Valid methods include GET, PUT, POST, DELETE, PATCH, and HEAD.

Max Response Time

Min Response Time

Port

Process ID

Service Name

Service ID The service ID isn't displayed in the Tivoli Enterprise Portal or the OMEGAMON enhanced 3270 user interface.

SMF ID

Request Count

JVM Extended z/OS Connect Request Details attributes

Managed System

SMF ID

Collector ID

Job Name

ASID

Process ID

Event Time

Event Time Ext

Timeout

API Name

Request URI

Method Possible options include HEAD, GET, PUT, DELETE and OPTIONS.

Query String

HTTP Code

Request ID

Total Request Time

zOS Connect Time

Remote Address

Request JSON Length

Response JSON Length

Request JSON

Response JSON

Service Name

Request Type

SoR ID

SoR Ref

SoR Resource

SoR Resp Time

Correlator

Operation

Provider

User ID

User Name

Mapped User ID

Port

JVM Extended z/OS Connect Request Summary attributes

Managed System

SMF_ID

Collector ID

Job Name

ASID

Process ID

Interval Time

Request Count

API Name

METHOD

API METHOD

Request URI

SoR ID

SoR Ref

SoR Resource

Resp Time Min

Resp Time Max

Resp Time Avg

Resp Time StdDev

Avg Request Length

Avg Response Length

Service Name

SoR Time Min

SoR Time Max

SoR Time Avg

SoR Time StdDev

zOSConnect Min

zOSConnect Max

zOSConnect Avg

zOSConnect StdDev

Error Count

Timeout Count

Situations

IBM Z OMEGAMON for JVM (IBM Z OMEGAMON for JVM) provides predefined situations that monitor for conditions in your Java virtual machine environments on z/OS that are typically considered to be problematic or noteworthy. When triggered, the situations generate Critical or Warning events in the Tivoli Enterprise Portal. Each situation provides expert advice for handling the condition if it arises.

The predefined situations that are provided with IBM Z OMEGAMON for JVM are not set to autostart. You must activate them before they can begin monitoring. Before you activate the situations, evaluate them and determine which ones you want to use and whether the conditions are appropriate for your environment. Using the Situation Editor in the Tivoli Enterprise Portal, you can change the conditions, relational operators, and comparison values to ones that are more appropriate for your environment. However, do not change the original predefined situations. Instead, copy the ones that you want to change, then make your changes to the copies.

IBM Z OMEGAMON for JVM includes the following categories of predefined situations:

CPU Statistics situations

The predefined situations in this category monitor Java virtual machine (JVM) processor conditions.

JVM_General_CPU_Critical

This critical alert triggers when the percentage of JVM processing time that occurs on general-purpose processors exceeds 75%.

The following formula is used: `JVM_CPU.General_CPU_% > 75.00`

JVM_General_CPU_Warning

This warning alert triggers when the percentage of JVM processing time that occurs on general-purpose processors exceeds 50%.

The following formula is used: `JVM_CPU.General_CPU_% > 50.00`

JVM_IFA_on_General_CPU_Critical

This critical alert triggers when the percentage of JVM processing time that occurs on general-purpose processors that was eligible to run on IFA processors, which were not available, exceeds 50%.

The following formula is used: `JVM_CPU.IFA_on_General_CPU_% > 50.00`

JVM_IFA_on_General_CPU_Warning

This warning alert triggers when the percentage of JVM processing time that occurs on general-purpose processors that was eligible to run on IFA processors, which were not available, exceeds 25%.

The following formula is used: `JVM_CPU.IFA_on_General_CPU_% > 25.00`

JVM Garbage Collection situations

The predefined situations in this category monitor Java virtual machine (JVM) garbage collection conditions.

JVM_GC_Pause_Critical

This critical alert triggers when the percentage of time in garbage collection pauses is more than 5.

The following formula is used: `JVM_Garbage_Collection.%_Time_in_GC_Pauses > 5.00`

JVM_GC_Pause_Warning

This warning alert triggers when the percentage of time in garbage collection pauses is more than 2.

The following formula is used: `JVM_Garbage_Collection.%_Time_in_GC_Pauses > 2.00`

JVM_Occupancy_after_GC_Critical

This critical alert triggers when the average occupancy percentage is more than 80.

The following formula is used: `JVM_Garbage_Collection.Avg_Occupancy_% > 80.00`

JVM_System_GC_Warning

This warning alert triggers if a garbage collection was forced by the system during the garbage collection interval.

The following formula is used: `JVM_Garbage_Collection.System_GC_Count >= 1`

Health Summary situations

The predefined situations in this category monitor Java virtual machine (JVM) lock and thread conditions.

JVM_Lock_Missed_Critical

This critical alert triggers when JVM threads wait to enter a lock on synchronized code for more than 50% of the total attempts.

The following formula is used: `JVM_Locks.Missed_% > 50.00`

JVM_Lock_Missed_Warning

This warning alert triggers when JVM threads wait to enter a lock on synchronized code for more than 10% of the total attempts.

The following formula is used: `JVM_Locks.Missed_% > 10.00`

JVM_Thread_Blocked_Warning

This warning alert triggers if a JVM thread is blocked.

The following formula is used: `JVM_Threads.State = BLOCKED`

z/OS Native Memory situations

The predefined situations in this category monitor z/OS-managed virtual storage availability within the JVM address space.

JVM_MEM24Free_Warning

This warning alert triggers when available 24-bit memory falls below 20%.

The following formula is used:

```
If the value of ZOSMEM.Mem24_Free_Pct is less than 20
and the value of ZOSMEM.Mem24_Free_Pct is greater than or equal to 10
then situation JVM_MEM24Free_Warning is true
```

JVM_MEM24Free_Critical

This critical alert triggers when available 24-bit memory falls below 10%.

The following formula is used:

```
If the value of ZOSMEM.Mem24_Free_Pct is less than 10
then situation JVM_MEM24Free_Critical is true
```

JVM_MEM31Free_Warning

This warning alert triggers when available 31-bit memory falls below 20%.

The following formula is used:

```
If the value of ZOSMEM.Mem31_Free_Pct is less than 20
and the value of ZOSMEM.Mem31_Free_Pct is greater than or equal to 10
then situation JVM_MEM31Free_Warning is true
```

JVM_MEM31Free_Critical

This critical alert triggers when available 31-bit memory falls below 10%.

The following formula is used:

```
If the value of ZOSMEM.Mem31_Free_Pct is less than 10
then situation JVM_MEM31Free_Critical is true
```

JVM_MEM64Free_Warning

This warning alert triggers when available 64-bit memory falls below 20%.

The following formula is used:

```
If the value of ZOSMEM.Mem64_Free_Pct is less than 20
and the value of ZOSMEM.Mem64_Free_Pct is greater than or equal to 10
then situation JVM_MEM64Free_Warning is true
```

JVM_MEM64Free_Criticale

This critical alert triggers when available 64-bit memory falls below 10%.

The following formula is used:

```
If the value of ZOSMEM.Mem64_Free_Pct is less than 10
then situation JVM_MEM64Free_Critical is true
```

z/OS Connect Request Summary situations

The predefined situations in this category monitor z/OS Connect Request conditions.

JVM_Avg_Elapsed_Time_Warning

This warning alert triggers if average elapsed time exceeds 1 second.

The following formula is used: `JVM_Avg_Elapsed_Time_in_Seconds ≥ 1`

z/OS Connect Extended Request situations

The predefined situations in this category monitor Java virtual machine (JVM) processor conditions.

JVM_zOS_Connect__Http_Warning

A z/OS Connect request completed with an HTTP status code greater than 299.

The following formula is used:

```
*IF *VALUE JVM_zOSConnect_Details.HTTP_Code *GT 299
```

JVM_zOS_Connect_Resp_Warning

A z/OS Connect request was found with moderately high total request response time. The fields `zOS_Conn_Time` and `SoR_Resp_Time` will tell you whether the high response time occurred in z/OS Connect or the System of Record.

The following formula is used:

```
*IF *VALUE JVM_zOSConnect_Details.Total_Request_Time
*GT 10.000000 *AND *VALUE
JVM_zOSConnect_Details.Total_Request_Time *LE 20.000000
```

JVM_zOS_Connect_Resp_Critical

A z/OS Connect request was found with very high total response time. The fields `zOS_Conn_Time` and `SoR_Resp_Time` will tell you whether the high response time occurred in z/OS Connect or the System of Record.

The following formula is used:

```
*IF *VALUE JVM_zOSConnect_Details.Total_Request_Time *GT 20.000000
```

JVM_zOS_Connect_Timeout_Warning

An asynchronous request timeout occurred. The HTTP Code may return 200 as the work completed successfully in the system of record, but if the asynchronous request timeout expired before the request had finished processing, the HTTP client will have actually received the HTTP status code 503

The following formula is used:

```
*IF *VALUE JVM_zOSConnect_Details.Timeout *EQ Y
```

Index

A

attributes [1](#)

E

enhanced 3270 user interface [1](#)
enhanced 3270UI [2](#)

J

JVM Auto Discovery attributes [14](#)
JVM Completed Job Detail attributes [15](#)
JVM Completed Job Summary attributes [15](#)
JVM CPU attributes [16](#)
JVM Environment Data attributes [16](#)
JVM Environment Data Details attributes [17](#)
JVM Garbage Collection attributes [17](#)
JVM Health Summary attributes [18](#)
JVM Locks attributes [19](#)
JVM Native Memory attributes [20](#)
JVM Overview attributes [20](#)
JVM Request Details attributes [22](#), [24](#)
JVM Request Summary attributes [23](#), [25](#)
JVM Threads attributes [21](#)
JVM Virtual Memory attributes [22](#)

M

monitoring resources [1](#)

O

OMEGAMON enhanced 3270 user interface [2](#)

R

resources
 monitoring [1](#)

S

situations [1](#)

T

Tivoli Enterprise Portal [1](#), [2](#)

U

user interface
 enhanced 3270UI [2](#)
 Tivoli Enterprise Portal [2](#)

W

workspaces
 Address Space Memory [3](#)
 Completed Job Detail [4](#)
 Completed Job Summary [3](#)
 CPU Statistics [4](#)
 Environment Data [4](#)
 Garbage Collection Details [5](#)
 Garbage Collection Statistics [6](#)
 Health Summary [6](#)
 JVM Overview [7](#)
 Lock Statistics [8](#)
 Native Memory [8](#)
 Thread Statistics [9](#)
 z/OS Connect Request Details [9](#), [10](#)
 z/OS Connect Request Summary [10](#), [12](#)
 z/OS Connect Slowest Requests [9](#)

