



BEA WebLogic Server™®

Using the Performance Monitor



Copyright

Copyright © 2004 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.



Contents

Getting Started With the Performance Monitor.....	1-1
Overview	1-1
Installing the Performance Monitor	1-2
Installing the Administration Server Component	1-3
Installing the Agent Component	1-4
Getting Started	1-6
Starting the Performance Monitor	1-6
About the Performance Monitor	1-8
Using the Performance Monitor.....	2-1
Overview	2-1
Agent Technology	2-2
Log File Monitor (dsLog)	2-3
Operating System Monitor (dsSys)	2-3
Network Service Test (dsProto)	2-3
Application Server (dsAppSrv)	2-3
Console Extension	2-4
Performance Monitor Web Application	2-4
About System Resource Monitoring	2-5
About Statistics Monitoring	2-6
Tasks	2-7
Viewing Alerts	2-7
Turning On E-mail Notification	2-7
Starting Log File Monitoring	2-8

Viewing Log File Monitoring	2-9
Log Monitor Configuration	2-10
Starting Network Service Tests	2-12
Viewing Network Service Tests	2-12
Network Service Test Configuration	2-13
Starting Machine System Resource Monitoring	2-14
Viewing Machine System Resource Monitoring	2-14
Starting Threshold Monitoring	2-15
Viewing Threshold Monitoring	2-16
Threshold Configuration	2-17
Filtering Results	2-19
Starting Statistics Monitoring	2-20
Viewing Statistics Monitoring	2-21
Stopping Monitoring	2-22
Working with Data Charts	2-22
Starting JMX Application Monitoring	2-23
Viewing JMX Application Monitoring	2-25
JMX Monitor Configuration	2-25
JMX Application Monitoring Example	2-26
Starting Application Monitoring	2-28
Stopping Application Monitoring	2-29
Viewing Application Monitoring	2-29
Starting Instrumentation	2-30
Viewing Method Statistics	2-33
Stopping Tracing	2-34
Instrumentation Example	2-35
Troubleshooting the Performance Monitor	3-1

Performance Monitor --> Resources --> Machine --> Process List4-1
Performance Monitor --> Configuration --> Instrumentation5-1
Performance Monitor --> Configuration --> JMX6-1
Performance Monitor --> Configuration --> Log7-1
Performance Monitor --> Configuration --> Network Service Test8-1
Performance Monitor --> Configuration --> Notifications9-1
Performance Monitor --> Configuration --> Thresholds10-1
Performance Monitor --> Resources --> Alerts11-1
Performance Monitor --> Resources --> Logs12-1
Performance Monitor --> Resources --> Machine --> Disks --> File Sys-
tems13-1
Performance Monitor --> Resources --> Machine --> Disks --> Physical
Disks14-1
Performance Monitor --> Resources --> Machine --> Memory/Swap
15-1
Performance Monitor --> Configuration --> Agent16-1
Performance Monitor --> Resources --> Machine --> Processor17-1
Performance Monitor --> Resources --> Network Service Test18-1
Performance Monitor --> Resources --> Thresholds19-1
Performance Monitor --> Statistics --> Applications20-1
Performance Monitor --> Statistics --> EJB Modules21-1
Performance Monitor --> Statistics --> JDBC22-1
Performance Monitor --> Statistics --> JMS --> Destinations23-1
Performance Monitor --> Statistics --> JMS --> Durables24-1
Performance Monitor --> Statistics --> JMS --> Servers25-1
Performance Monitor --> Statistics --> JMX26-1
Performance Monitor --> Statistics --> Servers --> Connections27-1
Performance Monitor --> Statistics --> Servers --> Execute Queue28-1
Performance Monitor --> Statistics --> Servers --> Heap29-1
Performance Monitor --> Statistics --> Servers --> JTA30-1
Performance Monitor --> Statistics --> Servers --> Sockets31-1
Performance Monitor --> Statistics --> Unassociated32-1
Performance Monitor --> Statistics --> Web Applications33-1
Performance Monitor --> Statistics --> Application Module --> EJBs
34-1

Performance Monitor --> Statistics --> Application Module --> JSPs
35-1

Performance Monitor --> Statistics --> Application Module --> Servlets
36-1

Performance Monitor --> Statistics --> Application Module --> Instru-
mented Classes37-1

1 Getting Started With the Performance Monitor

Overview

The Performance Monitor is a utility you use to monitor the performance of a WebLogic Server domain. You can use the Performance Monitor to diagnose and troubleshoot problems before they affect the performance of your applications. The utility can also alert you by email to adverse conditions within your WebLogic Server environment.

The Monitoring Extension can be installed at any time on a WebLogic Server domain. The installation consists of two parts:

- Installing the monitoring agent on desired server instances.
- Installing the interface views used to configure and view performance monitoring.

The Performance Monitor is a very economical utility, and requires only a very small amount of system resources.

Installing the Performance Monitor

There are two components that need to be installed for the Performance Monitor. One component, the Performance Monitor interface, needs to be installed on the Administration Server. See [“Installing the Administration Server Component”](#) on page 1-3 for complete details.

The other component, the agent, needs to be installed on each physical machine hosting a WebLogic Server instance that you want to monitor. See [“Installing the Agent Component”](#) on page 1-4 for complete details.

Figure 1-1 Architecture Diagram for a Single Machine Cluster

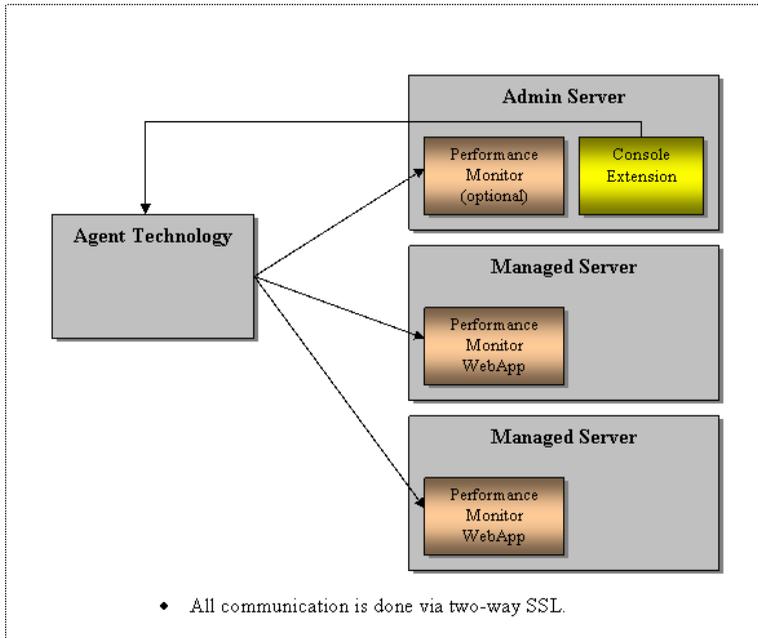
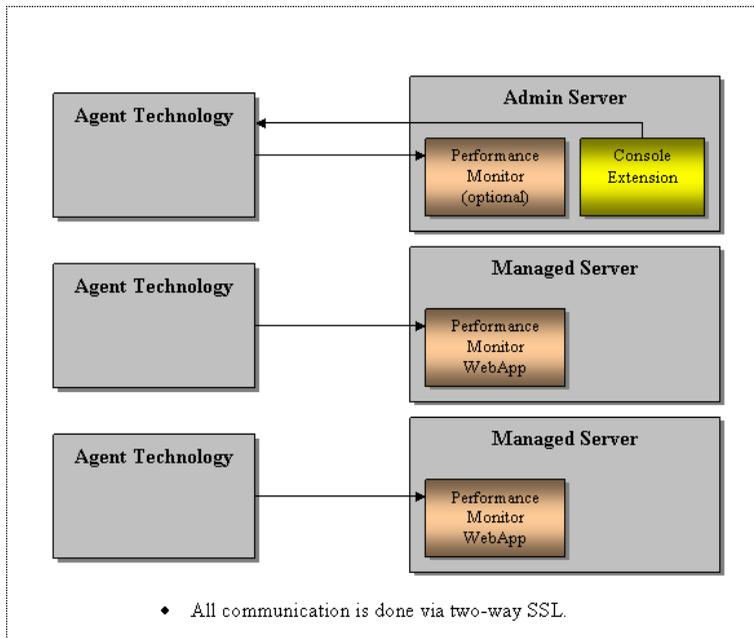


Figure 1-2 Architecture Diagram for a Multiple Machine Cluster



Installing the Administration Server Component

The Administration Server component includes the views required to configure and use the Performance Monitor. Install this component on the machine hosting the Administration Server.

1. 1. Extract all files from the Performance Monitor *.tar file, and run the `setup.exe/sh` file. The Performance Monitor installation view opens.

Note: The Administration Server Component must be installed as the same user/role as the WebLogic Server instance (Unix and Windows).
2. **License Agreement:** Read thoroughly, and click Next to continue.
3. **WebLogic Home Directory:** Enter the directory path for WebLogic Home, for example, `c:\bea\weblogic81`. Use Browse, if necessary, to navigate to the directory.

Note: You must install the Performance Monitor component in the WebLogic Home directory you specified during installation of WebLogic Server. By default, this directory is named `weblogic81`.

Once the WebLogic Home directory is entered, click Next to continue.

4. **Package Selection:** Select **Platform Independent**.

Note: If a WebLogic Server instance is configured on this computer, and you would like to monitor this server, select **Agent Technology** also. See [“Installing the Agent Component” on page 1-4](#) for information on Agent Technology install dialog boxes.

5. **AGENT_PORT:** Enter the port that the Performance Monitor agent will run on for communication with WebLogic Server. By default the port is 8080. Once the agent port is entered, click OK to continue.

Note: The Agent Port needs to be set the same for the Administration Server component and all Agent components as well.

6. **Use Secure HTTP:** Click Yes if you would like to use Secure HTTP to communicate with the Performance Monitor Agents.

Note: You must select the same answer for the Administration Server component and all Agent components.

7. **Package Progress:** Displays progress of the Performance Monitor installation.

8. **Installation Complete:** Shows the results of the installation. Click View Install Log to view any errors, or click Finish to complete the installation.

Next, install the agent component on each server instance you want to monitor. See [“Installing the Agent Component” on page 1-4](#).

Installing the Agent Component

The Agent component runs as an external service or daemon. It retrieves and stores performance statistics from applications and resources running on the server. Install this component on each server you want to monitor.

1. Extract all files from the Performance Monitor `*.tar` file, and run the `setup.<exe/sh>` file. The Performance Monitor installation view opens.

Note: On Solaris systems, run the setup program as root. After installation, the Agent must also run as root. On Windows systems, if you plan to install the Agent as a Windows Service, run the setup program with Administrator privileges.

2. **License Agreement:** Read thoroughly, and click Next to continue.
3. **WebLogic Home Directory:** Enter the directory path for WebLogic Home (ex. c:\bea\weblogic81). Use Browse, if necessary, to navigate to the directory.

Note: You must install the Performance Monitor component in the WebLogic Home directory you specified during installation of WebLogic Server. By default, this directory is named `weblogic81`.

Once the WebLogic Home directory is entered, click Next to continue.

4. **Package Selection:** Select **Agent Technology**.
5. **AGENT_PORT:** Enter the port that the Performance Monitor agent will run on for communication with WebLogic Server. By default the port is 8080. Once the agent port is entered, click OK to continue.

Note: The Agent Port needs to be set the same for the Administration Server component and all Agent components as well.

6. **Use Secure HTTP:** Click Yes if you would like to use Secure HTTP to communicate with the Performance Monitor Agents.

Note: You must select the same answer for the Administration Server component and all Agent components.

7. **Install As Service** (Windows installations only): Click Yes if you want to install the agent as a Windows service. This enables you to start and stop the agent through the Services console in the Windows Administrative Tools. You can also configure the agent to start automatically when the computer is restarted.
8. **Install Menu Options** (Windows installations only): Click Yes if you would like to add the Start Agent, Stop Agent, and Uninstall shortcuts to the Windows Programs menu.
9. **Package Progress:** Once the installation is complete, click Finish.
10. **Installation Complete:** Shows the results of the installation. Click View Install Log to view errors if any occurred, or click Finish to complete the installation.

Repeat this procedure on each server instance that you want to monitor. Proceed to [“Getting Started” on page 1-6](#) for the next steps required to configure and start the Performance Monitor.

Getting Started

Starting the Performance Monitor

Once the Performance Monitor is installed, these three steps are required to configure and start it:

- [“Starting the Agent” on page 1-6](#)
- [“Modifying the WebLogic Start Script” on page 1-7](#)
- [“Deploying the Performance Monitor Web Application” on page 1-7](#)

Starting the Agent

Start the Performance Monitor agent on each server instance you want to monitor.

1. If you are using the Windows operating system, and chose to install the Performance Monitor shortcuts to the Windows Programs menu, you can start the agent by clicking Start > Programs > BEA Performance Monitor > Start Agent.

If you chose to install the agent as a Windows service, you can use the Services console to start and stop the agent. You can also configure the agent to start automatically when the computer is restarted.

Note: If you did not install the agent as a Windows service but would like to do so, simply reinstall the agent on each server instance as shown in [“Installing the Agent Component” on page 1-4](#).

2. For both UNIX-based and Windows operating systems, you can start the agent from a command prompt. Change to the `$WL_HOME/common/perf/bin` directory. Enter this command:

```
dsAgent -console
```

UNIX start scripts are installed in the `$WL_HOME/common/perf/bin` directory. To automatically start the agent when the computer is restarted, see [“Installing the Agent as a Unix Daemon” on page 1-8](#).

Proceed to [“Modifying the WebLogic Start Script” on page 1-7](#).

Modifying the WebLogic Start Script

Modifying the WebLogic start script enables the Performance Monitor’s instrumentation feature. Instrumentation can retrieve response times, number of calls (iterations), and exception counts for methods operating within a component. This feature offers an in-depth look at individual method performance within a component.

1. To enable instrumentation, locate the server start script and open it in a text editor. The start script is usually located here:

```
.../<WL_Home>/<WL_Server_domain>/start<WL_domain>.<cmd/sh>
```

2. Enter the following parameter in the java process line anywhere after the classpath entry and before `weblogic.Server`:

```
-Dweblogic.classloader.preprocessor=com.dirig.preprocessor.  
DirigBEAClassProcessor
```

3. Save the start script, and restart WebLogic Server.

Proceed to [“Deploying the Performance Monitor Web Application” on page 1-7](#).

Deploying the Performance Monitor Web Application

Deploying the Performance Monitor web application enables the agent to retrieve JMX MBean statistics from applications deployed by WebLogic Server. Any deployed application that uses MBeans, including enterprise applications (EARs), can be monitored by the Performance Monitor agent.

1. In the Administration Console’s left panel, click Performance Monitor. In the right panel, select [click here](#).
2. In the right panel, click the Deployments tab.
3. The Independent Servers table displays a list of servers configured for this domain. Check the box next to each server that you want to monitor, and click Apply.

The Performance Monitor web application should be displayed in the Deployments > Web Application Modules node in the Administration Console. To learn more about configuring and using the Performance Monitor, proceed to [“Getting Started” on page 1-6.](#)

Installing the Agent as a Unix Daemon

To set up the pep agent as a unix Daemon you will need to copy the startup script over to the `init.d` directory and use the Unix symbolic link feature to attach them to the `rc2.d` directories.

Use the commands below specific for your version of Unix. These commands will need to be run as root.

```
# For SunOS & SunOS64
cp $WL_HOME/common/perf/bin/initdsAgent.sh /etc/init.d/initdsAgent
ln -s /etc/init.d/initdsAgent /etc/rc2.d/S999dsAgent
ln -s /etc/init.d/initdsAgent /etc/rc2.d/K999dsAgent

# For Linux
cp $WL_HOME/common/perf/bin/initdsAgent.sh
  /etc/rc.d/init.d/initdsAgent
ln -s /etc/rc.d/init.d/initdsAgent /etc/rc.d/rc2.d/S999dsAgent
ln -s /etc/rc.d/init.d/initdsAgent /etc/rc.d/rc2.d/K999dsAgent

# For HPUX64
cp $WL_HOME/common/perf/bin/initdsAgent.sh
  /sbin/init.d/initdsAgent
ln -s /sbin/init.d/initdsAgent /sbin/rc2.d/S999dsAgent
ln -s /sbin/init.d/initdsAgent /sbin/rc2.d/K999dsAgent

# For AIX
cp $WL_HOME/common/perf/bin/initdsAgent.sh /etc/rc.dsAgent
mkitab dsAgent:2:once:/etc/rc.dsAgent
```

About the Performance Monitor

Once the Performance Monitor is installed and started, it can be configured to monitor system resources (“Resources”) and applications (“Statistics”) within the WebLogic environment.

The Performance Monitor agent retrieves statistics from system and application attributes. Most statistics are for display purposes, and can be viewed by system administrators. Other statistics are compared to Performance Monitor rules to see if they are within acceptable operating parameters. If a data value falls outside these parameters, the Performance Monitor will automatically notify the system administrator by generating an alert.

Alerts are used to notify the administrator of adverse conditions detected by the Performance Monitor. Any violations of Performance Monitor rules result in the generation of alerts. Alerts are classified by severity (low, medium, and high). Alerts are configured for events that range from minor incidents that won't affect system performance, such as a specific log entry occurrence, to major events that will impact server performance, such as too much of the JVM heap being consumed.

Note: It is difficult to recommend a standard configuration for the Performance Monitor due to the customized nature of WebLogic Server environments. Users may be interested in different features, so optimum configuration usually results from experimentation to determine what statistics are of most interest.

The following sections contain information about specific features. See [“Using the Performance Monitor” on page 2-1](#) for complete details on configuring and viewing performance monitoring.

About Resources

The Resources feature of the Performance Monitor includes the following capabilities:

- It monitors performance statistics for CPU, memory, swap, and file system capacity. You can visually monitor machine statistics in the Performance Monitor views, and take action if resource consumption is too high. Excessive resource consumption can also be an initial indicator of adverse conditions within WebLogic Server environment.

See [“Starting Machine System Resource Monitoring” on page 2-14](#) for more information.
- Log files can be monitored for defined error messages. These error messages are excellent indicators of adverse conditions in the WebLogic environment, and can help administrators hone in on the source of the problem.

See [“Starting Log File Monitoring” on page 2-8](#) for more information.

-
- Network service tests are performed to check connectivity with other computers in the network. The response time of the test is monitored; higher response times can indicate problems.

See [“Starting Network Service Tests” on page 2-12](#) for more information.

- Threshold monitoring works in conjunction with machine system resource, statistics, and network service monitoring. Thresholds set acceptable operating limits for these statistics, and generates an alert if statistics exceed these limits.

See [“Starting Threshold Monitoring” on page 2-15](#) for more information.

About WebLogic Server Resource Monitoring

The Statistics feature monitors many aspects of the WebLogic Server environment, such as server resource allocation and application performance. Administrators using Statistics monitoring should be familiar with the WebLogic Server environment and its components.

Statistics monitoring, like Resource monitoring, can be very useful for discovering problems and honing in on their source. For example, if the messaging service has too many messages pending, or if all connection pools are in use, the administrator can take appropriate action. Most data retrieved from the Statistics monitor is for display purposes, with the exception of JVM heap utilization (a threshold is configured for this).

See [“Starting Statistics Monitoring” on page 2-20](#) for more information.

About Java Management Extensions (JMX) Monitoring

The JMX monitor can retrieve performance data from MBeans associated with applications deployed by WebLogic Server. This is extremely useful for monitoring the application’s MBean performance metrics, called attributes. Administrators using JMX monitoring should be familiar with the JMX application and its MBeans. Knowing the program code for the application is not absolutely necessary, but it is necessary to know which MBean attributes are exposed, and what they represent. It may be helpful to consult with the application developer prior to using JMX monitoring.

This feature is extremely useful for monitoring specific applications, and pinpointing explicit problems within those applications. Administrators can monitor MBean attributes with integer values, such as counters and response times, and attributes with

string values, such as a state (ex. on/off). If a numerical attribute value is excessive, or a string attribute value changes unexpectedly, the administrator can take appropriate action.

See [“Starting JMX Application Monitoring” on page 2-23](#) for more information. Refer also to [“JMX Application Monitoring Example” on page 2-26](#) to see an example of JMX MBean monitoring.

About Application Monitoring

The Applications monitor can retrieve performance data from applications deployed on the WebLogic server. Applications include enterprise applications (contained in EAR files), web applications (contained in WAR files), and EJB Modules (contained in JAR files).

Application monitoring retrieves statistics such as number of requests and response times for components such as servlets, EJBs, and JSPs. This is useful for not only indicating potential problems, but also identifying the source right down to the application component.

See [“Starting Application Monitoring” on page 2-28](#) for more information. For a more in-depth view of application performance, refer to the following section.

About Instrumentation

In conjunction with application monitoring, the Performance Monitor has an instrumentation feature that allows users to monitor individual methods operating within an application. Instrumentation can retrieve method performance statistics such as iteration, response time, and exceptions.

This is extremely useful for providing an in-depth look at application performance, and for pinpointing exact problems within an application. In order to use instrumentation effectively, administrators should have a good working knowledge of the application--and its methods--to be monitored.

See [“Starting Instrumentation” on page 2-30](#) for more information. Refer also to [“Instrumentation Example” on page 2-35](#) to see an example using instrumentation to monitor method performance.



2 Using the Performance Monitor

Overview

The Performance Monitor is a utility you use to monitor the performance of a WebLogic Server domain. You can use the Performance Monitor to diagnose and troubleshoot problems before they affect the performance of your applications. The utility can also alert you by email to adverse conditions within your WebLogic Server environment.

The Performance Monitor—>Summary page displays a brief view of performance monitoring configuration. You can use the links in this view to go directly to specific pages within the Performance Monitor.

The Performance Monitor node in the Administration Console contains three sub-nodes:

- Resources: Displays the results of system resource monitoring.
- Statistics: Displays the results of monitoring WebLogic Server resources and applications.
- Configuration: Contains the interfaces used to configure the Performance Monitor.

To learn more about configuring and using the Performance Monitor, refer to the following sections:

[“About System Resource Monitoring” on page 2-5](#)

[“About Statistics Monitoring” on page 2-6](#)

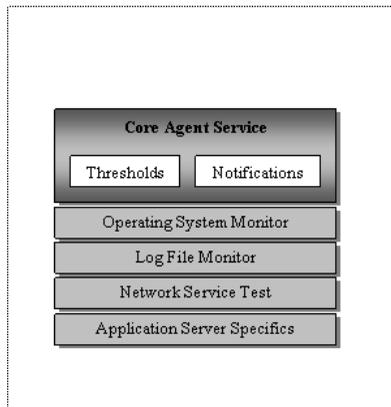
Agent Technology

The agent is a process external to a WebLogic Server instance. It runs as a Windows Service or Unix daemon. Its main purpose is to store statistical information for threshold validation as well as send notifications when a statistic has exceeded a boundary. It is comprised of four subsystems that are described below.

The Agent technology uses SSL to communicate with WebLogic server. In order to accomplish this, a certificate is needed by the agent process. There are two ways to get a valid certificate. By default, the agent will convert the default WebLogic certificate (JKS format) into a format it can read (PEM format). See [“I seem to be experiencing certificate conflicts.” on page 3-3](#) in the troubleshooting guide for more details. If you would like to dedicate your own certificate, edit the

`$WL_HOME/common/perf/perf.properties` file. For information about the format of this file, and instructions on editing it, see [Node Manager Properties](#).

Figure 2-1 Agent Technology



Log File Monitor (dsLog)

The log monitor subagent watches system, security, application log files and any ASCII file for patterns. These patterns are a series of text messages, which are possible entries in the log.

Operating System Monitor (dsSys)

The operating system monitor is responsible for monitoring a system's CPU, memory, physical disks and file systems. This subsystem collects different statistics on these components.

Network Service Test (dsProto)

The Network Service Test subsystem allows users to determine the availability and response time for the most common services, Email and Web Server. Using its simulated transaction engine, the Agent will emulate certain protocols, such as HTTP/S, FTP, SMTP, DNS, POP3, ICMP Ping and TCP/IP ECHO. During emulation, the agent reports the connection time, last transaction time, and average transaction time of each operation. For example, using the HTTP protocol, the agent will connect to a Web Server and record the connection interval. After connect, the agent will attempt a download of a specific web page. If successful, the round-trip transaction time is stored, and averaged in two separate statistics. Optionally, a corrective action can be taken if the protocol emulation failed, or the initial connect has failed.

Application Server (dsAppSrv)

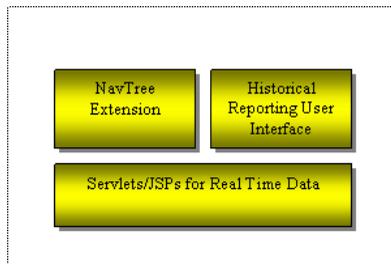
The application server subsystem communicates with the Performance Monitor Web Application to retrieve the latest application server statistics. These statistics can either come from the MBean server or the instrumentation interface within the Performance Monitor Web application.

Console Extension

The console extension is used to create the tree items found in the console Navigation Tree (left side). It is automatically deployed from the `$WL_HOME/common/perf` directory, provided that the `PerformanceMonitor.war` file exists in the `$WL_HOME/common/perf/java` directory.

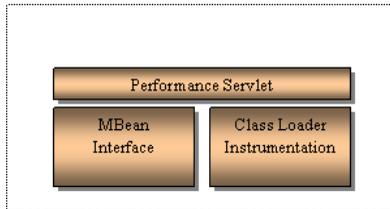
The console extension displays historical data and allows for basic configuration changes as well as live statistical reporting. All of this information is stored in the agent, and displayed within the console extension servlets and JSPs.

Figure 2-2 Console Extension



Performance Monitor Web Application

The Performance Monitor Web application is deployed to each monitored server instance, and is used to consolidate statistics from that server instance and any applications and/or services deployed within it. Large portions of the statistics gathered by the Performance Monitor Web application are accessed via the MBean server, with the exception of instrumented classes. Clicking on the `PerformanceMonitor` folder within the WLS console Navigation Tree deploys this web application.

Figure 2-3 Performance Monitor Web Application

About System Resource Monitoring

The Performance Monitor offers the ability to monitor operating system resources, such as:

- CPU capacity
- Memory capacity
- Swap capacity
- File System utilization

The Performance Monitor can also scan WebLogic Server log files for entries that indicate errors or events. Network connectivity and response times can also be monitored.

To learn more about resource monitoring, refer to the following sections:

[“Performance Monitor --> Resources --> Logs”](#) on page 12-1

[“Performance Monitor --> Resources --> Network Service Test”](#) on page 18-1

[“Performance Monitor --> Resources --> Machine --> Processor”](#) on page 17-1

[“Performance Monitor --> Resources --> Machine --> Memory/Swap”](#) on page 15-1

[“Performance Monitor --> Resources --> Machine --> Disks --> File Systems”](#) on page 13-1

[“Performance Monitor --> Resources --> Machine --> Disks --> Physical Disks”](#) on page 14-1

[“Performance Monitor --> Resources --> Machine --> Process List”](#) on page 4-1

[“Performance Monitor --> Resources --> Thresholds” on page 19-1](#)

About Statistics Monitoring

The Performance Monitor offers the ability to monitor many WebLogic Server resources such as connection pool, heap, and JMS messaging performance. You can also monitor statistics (such as number of calls and response times) for deployed applications, JMX objects, or modules.

The Performance Monitor also includes a tool called Instrumentation. Instrumentation enables you to delve deeply into an application or module, and monitor the performance of each method operating within the parent object. Not only can you monitor the performance of objects such as servlets and EJBs, but you can use instrumentation to identify and monitor the performance of their methods. This is very useful for finding the precise cause of problems within servers.

To learn more about statistics monitoring, refer to the following sections:

[“Performance Monitor --> Statistics --> Servers --> Heap” on page 29-1](#)

[“Performance Monitor --> Statistics --> Servers --> Execute Queue” on page 28-1](#)

[“Performance Monitor --> Statistics --> Servers --> JTA” on page 30-1](#)

[“Performance Monitor --> Statistics --> Servers --> Sockets” on page 31-1](#)

[“Performance Monitor --> Statistics --> Servers --> Connections” on page 27-1](#)

[“Performance Monitor --> Statistics --> JDBC” on page 22-1](#)

[“Performance Monitor --> Statistics --> JMS --> Servers” on page 25-1](#)

[“Performance Monitor --> Statistics --> JMS --> Destinations” on page 23-1](#)

[“Performance Monitor --> Statistics --> JMS --> Durables” on page 24-1](#)

[“Performance Monitor --> Statistics --> JMX” on page 26-1](#)

[“Performance Monitor --> Statistics --> Applications” on page 20-1](#)

[“Performance Monitor --> Statistics --> EJB Modules” on page 21-1](#)

[“Performance Monitor --> Statistics --> Web Applications” on page 33-1](#)

[“Performance Monitor --> Statistics --> Unassociated” on page 32-1](#)

Tasks

Viewing Alerts

To view existing alerts:

1. In the Administration Console, open the Performance Monitor folder.
2. Open the Resources folder. Select Alerts.
3. The Alerts view is displayed in the right panel, and provides the following information:
 - Agent: Specifies which machine the alert occurred on.
 - Time: Shows the time and date when the alert occurred.
 - Severity: Shows the severity level of the alert (low, medium, or high).
 - Text: Brief text message describing the alert.
4. You can sort the results in ascending or descending order by clicking any of the column headers. For example, clicking Severity will sort the results based on severity levels.

Turning On E-mail Notification

To configure the Performance Monitor to send an e-mail whenever an alert is generated:

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click Notifications. The Notification Configuration view is displayed in the right panel.

-
3. In the Email Configuration column, enter the following information for each Agent that you want to activate e-mail notification for:
 - On: Check this box to turn on notification.
 - Destination eMail: Enter the e-mail address that you want to receive the notification.
 - Source eMail: Enter an e-mail address that will be used as the sender's address, such as WLPerfMon@<yourcompany>.com.
 - SMTP Server: Enter the IP address or machine name for the SMTP mail server in your network. (This is a required field; do not leave blank.)
 4. Click Apply. You will now be notified via e-mail whenever an alert occurs.
 5. To turn off e-mail notification at any time, open the Notification Configuration view as shown previously. Uncheck the On box, and click Apply.

Starting Log File Monitoring

To begin monitoring WebLogic Server log file entries:

1. In the Administration Console, click the Performance Monitor folder.
2. Click the Configuration folder. Select Agent. The Agent Configuration table opens in the right panel.
3. The Default Ruleset is the set of monitoring rules that are supplied with the software. The log monitor ruleset specifies the regular expression patterns that the log monitor will look for in the log files. See [“Viewing Patterns for Log File Monitoring” on page 2-10](#) for more information.

In the Rules column, click Apply Default Ruleset to use the default ruleset supplied with the software.

If you've changed the log monitor configuration at any time, clicking Apply Default Ruleset will reset the log monitor rules to the default parameters. Any modifications made to the log monitor configuration will be lost.

4. In the Monitoring Options column, check the Log Monitoring box for each Agent that you want to monitor.

- Click Apply to begin monitoring log files. The log monitor will start scanning the log file for any new entries or modifications as they occur. See [“Viewing Log File Monitoring” on page 2-9](#).

Viewing Log File Monitoring

To view log file monitoring:

- In the Administration Console, click Performance Monitor.
- Click Resources, then select Logs.
- The Log Monitoring view is displayed in the right panel, and provides the following information:
 - Agent: Name of the host machine.
 - Name: Name of the log file being monitored.
 - Size (K): Size of the log file, in Kb.
 - Last Modified: The time and date that the log file was last modified.
 - Occurrences: Number of times that entries in the log file have matched the log monitor’s regular expression pattern.
- To view the actual log file entries, click the View Log icon as shown:

Agent	Name	Size (K)
50X	Server log: MedRecServer	56

The 50 most recent entries are displayed in a new window. To view the 50 entries prior to those, click the Previous 50 button.

- The log monitor ruleset scans log entries for several types of errors. When a pattern match occurs, you can see which type of error occurred by clicking on the number in the Occurrences column:

Agent	Name	Size (B)	Last Modified	Occurrences
SOX	Server log MedRecServer	56	Thu Apr 24 16:56:26 2003	1

The Performance Monitor —> Patterns view opens in a new window. Each log monitor pattern is listed, and the number of occurrences for each pattern is shown.

Note: In the main log monitoring view in the Administration Console, you can sort the results in ascending or descending order by clicking any of the column headers. For example, clicking Name will sort the results in alphabetical order based on the log file name.

When a pattern match occurs, an alert is generated. See [“Viewing Alerts” on page 2-7](#). You can also be notified immediately by e-mail when an alert occurs. See [“Turning On E-mail Notification” on page 2-7](#).

You can filter the results to show selected information. See [“Filtering Results” on page 2-19](#).

To make changes to the log monitor, see [“Log Monitor Configuration” on page 2-10](#).

Log Monitor Configuration

Viewing Patterns for Log File Monitoring

1. To see what regular expression patterns the log monitor is looking for, open the Performance Monitor —> Resources —> Logs view.
2. Click the number in the Occurrences field.



The Performance Monitor —> Patterns view lists the following patterns for the default ruleset:

- HTTP Error: Any errors with HTTP transmissions, such as a page load failure.
- EJB Error: Any errors related to EJBs.
- Security Error: Any security-related errors.
- WLS Error: Any WebLogic Server error messages.
- Deployer Error: Any errors related to the deployment of applications or modules.

Modifying Log Monitor Configuration

You can modify two parameters of the log monitor ruleset:

- The location of the log file.
- How frequently the log file is scanned for matching patterns.

To change log monitor configuration:

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click Log.
3. The Log Monitoring Configuration view is displayed in the right panel. In the Name column, click the name of the desired server.
4. The Edit Log view opens in a new window.
 - To change the location of the log file, or to specify another WebLogic Server log file name, manually enter the new file path in the File field.
 - The Interval field specifies how frequently the log monitor will scan the log file. To change the interval, manually enter the desired number in the Interval field. For units, use the drop-down menu to select Minutes or Hours.
5. Click OK to accept the changes. The changes will take effect immediately.

Starting Network Service Tests

To begin network service tests:

1. In the Administration Console, click the Performance Monitor folder.
2. Click the Configuration folder. Select Agent. The Agent Configuration table opens in the right panel.
3. The default ruleset is the set of monitoring rules that are configured within the software. See [“Network Service Test Configuration” on page 2-13](#) for more information.

In the Rules column, click Apply Default Ruleset to use the default ruleset supplied with the software.

If you’ve changed the configuration at any time, clicking Apply Default Ruleset will reset the ruleset to the default parameters. Any modifications made to the configuration will be lost.

4. In the Monitoring Options column, check the Network Service Test box for each Agent that you want to monitor.
5. Click Apply to start the Network Service Test. See [“Viewing Network Service Tests” on page 2-12](#).

Viewing Network Service Tests

To view the results of the network service tests:

1. In the Administration Console, click Performance Monitor.
2. Click Resources, then click Network Service Test.
3. The Network Service Test view is displayed in the right panel, and provides the following information:
 - Agent: Name of the host machine.
 - Name: Name of the network test.
 - Type: The protocol type of the network test.

- **Connect Time:** Amount of time, in milliseconds (ms), required to connect with the server being tested.
 - **Response Time:** Amount of time, in ms, required to successfully download the Web page from the remote server.
4. You can view graphs of historical data for the Connect and Response Times. In the Network Service Test view, click the number in the Connect Time or Response Time columns.

Type	Connect Time	Response Time
HTTP	0.000000	0.030000

The historical data chart opens in a new window. These data charts are useful for locating past trouble spots, and can help you discern performance patterns.

You can manipulate the data chart view to zoom in on segments of data, or to zoom out to see more data. See [“Working with Data Charts” on page 2-22](#).

Note: You can sort the results in the Network Service Test view in ascending or descending order by clicking on any of the column headers. For example, clicking Response Time will sort the results numerically based on the response time values.

You can filter the results to show selected information. See [“Filtering Results” on page 2-19](#).

If a network service test fails, you can be notified immediately by e-mail. See [“Turning On E-mail Notification” on page 2-7](#).

Network Service Test Configuration

You can view and modify the network service test configuration.

1. In the Administration Console, click the Performance Monitor folder.
2. Click Configuration, then click Network Service Test.

-
3. The Network Service Test Configuration view is displayed in the right panel. In the Name column, click the name of the desired test.
 4. The Edit Network Service Test view opens in a new window.
 - The Name and Network Service fields are not editable.
 - The Address field specifies the IP address or name of the server that will be tested. To change the server, enter the new IP address or name in the field.
 - The Interval specifies how frequently the network service test will attempt to download the Web page. To change the interval, manually enter the desired number in the Interval field. For units, use the drop-down menu to select Minutes or Hours.
 - The Page field specifies which Web page will be downloaded from the server. To change the web page, enter the correct file path in the field.
 5. Click OK to accept the changes. The changes will take effect immediately.

Starting Machine System Resource Monitoring

To begin monitoring machine system resources:

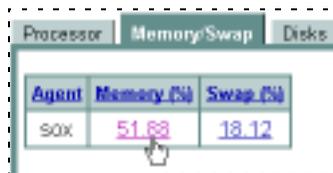
1. In the Administration Console, click the Performance Monitor folder.
2. Click the Configuration folder. Select Agent. The Agent Configuration table opens in the right panel.
3. In the Rules column, Apply Default Ruleset does not apply for machine system resource monitoring.
4. In the Monitoring Options column, check the Resources box for each Agent that you want to monitor.
5. Click Apply to start the Resources monitoring. See [“Viewing Machine System Resource Monitoring” on page 2-14](#).

Viewing Machine System Resource Monitoring

To view the results of the machine resource monitoring:

1. In the Administration Console, click Performance Monitor.
2. Click Resources, then click Machine.
3. The System Resources view is displayed in the right panel. Click the tabs to view various statistics for processor usage, memory/swap utilization, disk space allocation, and active processes.

If a statistic is displayed as a link, usually in a different color and underlined (depending on your browser configuration), you can click the number to open a historical data chart for that statistic.



Agent	Memory (%)	Swap (%)
SOX	51.88	18.12

The data chart will open in a new window. You can manipulate the chart view to zoom in on segments of data, or to zoom out to see more data. See [“Working with Data Charts”](#) on page 2-22.

Note: You can sort the results in the Resources view in ascending or descending order by clicking on any of the column headers. For example, clicking Name will sort the results in alphabetical order based on the resource name.

You can filter the results to show selected information. See [“Filtering Results”](#) on page 2-19.

Starting Threshold Monitoring

Threshold monitoring does not require starting, and occurs automatically when the appropriate monitor is started in the Agent Configuration.

For example, one of the thresholds is configured to monitor CPU usage. As soon as the resources monitor is started, the CPU threshold immediately starts comparing the CPU statistics to its parameters. If the resources monitor is stopped, the threshold will not report CPU threshold monitoring results.

See [“Threshold Configuration” on page 2-17](#) for complete information on which statistics thresholds are configured for.

Viewing Threshold Monitoring

To view the results of the threshold monitoring:

1. In the Administration Console, click Performance Monitor.
2. Click Resources, then click Thresholds.
3. The Thresholds view is displayed in the right panel, and provides the following information:
 - Agent: Name of the host machine.
 - Name: Name of the threshold.
 - State: Shows Sampling (establishing a baseline for comparing statistics) or Processing (comparing statistics to the baseline). See [“Threshold Configuration” on page 2-17](#) for information on baseline creation.
 - Base Value: Baseline value established by sampling.
 - Current Value: Most recent statistic retrieved by the Performance Monitor. Threshold rules are applied to this value, and then it’s compared to the base value. If the current value exceeds the threshold parameters, an alert is generated.

Note: You can sort the results in ascending or descending order by clicking on any of the column headers. For example, clicking Current Value will sort the results in numerical order based on the actual retrieved value of that statistic.

You can filter the results to show selected information. See [“Filtering Results” on page 2-19](#).

If a threshold generates an alert, you can be notified immediately by e-mail. See [“Turning On E-mail Notification” on page 2-7](#).

Threshold Configuration

Viewing Threshold Configuration

To view threshold configuration:

1. In the Administration Console, click the Performance Monitor.
2. Click Configuration, and then click Thresholds.
3. The Thresholds Configuration view is displayed. It lists each threshold, and the machine the threshold is monitoring.
4. To view the configuration of a threshold, click the name of the threshold in the Name column. The Edit Threshold view opens in a new window, and shows the values for these settings:
 - Name: Threshold name.
 - Trip Value: This is the amount that is applied to the base value to determine if a statistic is within the bounds of the threshold.
 - Operator: Percent Greater or Percent Less. This defines the units and the equation for the Trip Value. For example, a Trip Value of 25 and an Operator of Percent Greater means that the measured value must not exceed the base value by more than 25%.
 - Interval: This is how long the measured value must continuously violate the threshold parameters in order to generate an alert.
 - Severity: This is the severity level assigned to this threshold, and can be Low, Medium, or High.

The baseline value is automatically determined by the threshold. The threshold will monitor the attribute for three minutes, and retrieve—or *sample*—a value every 30 seconds. The average value of these sampled data points becomes the baseline value.

The baseline value is also a *rolling* value, and is recalculated periodically to accurately reflect the current performance values.

5. The following is a brief description of each threshold that comes with the Performance Monitor:

-
- **Slow Network Service Test:** The response time of the Network Service Test must not increase more than 25% over the baseline value for 5 consecutive minutes. It has a Low severity level.
The Network Service Test Monitor must be running; see [“Starting Network Service Tests” on page 2-12.](#)
 - **JVM Heap Utilization:** The percentage of JVM heap consumed must not increase more than 25% over the baseline value for 5 consecutive minutes. It has a Low severity level.
The Statistics Monitor must be running; see [“Starting Statistics Monitoring” on page 2-20.](#)
 - **CPU Threshold:** The percentage of CPU in use must not increase more than 25% over the baseline value for 5 consecutive minutes. It has a Low severity.
The Resources Monitor must be running; see [“Starting Machine System Resource Monitoring” on page 2-14.](#)
 - **Memory Threshold:** The percentage of memory in use must not increase more than 25% over the baseline value for 5 consecutive minutes. It has a Low severity.
The Resources Monitor must be running; see [“Starting Machine System Resource Monitoring” on page 2-14.](#)

Modifying Threshold Configuration

To change any of the threshold configuration values:

1. In the Administration Console, click the Performance Monitor.
2. Click Configuration, and then click Thresholds.
3. The Thresholds Configuration view is displayed. Click the name of the threshold in the Name column.
4. The Edit Threshold view opens in a new window. Modify the threshold values as desired. For complete information on threshold parameters, see [“Viewing Threshold Configuration” on page 2-17.](#)

Click OK. The changes will take effect immediately.

Filtering Results

To filter the results of system resource monitoring:

1. In the desired view, click Show Filter in the top right panel.
2. In the Agent field, click to select the Agent that you want to view.
3. In the Visible Rows field, enter the number of rows you wish to view. Click Apply Filter.

For example, if a user wanted to view the top three active processes based on memory consumption, the user would open the Resources —> Machine —> Process List tab. The user would then apply the filter to view the top three entries.

Once the filter was applied, the user would then click the Memory heading to sort the filtered list numerically. The view would display the top three processes based on memory consumption.

Performance Monitor > System Resources   

connected to : sox:7001 | welcome : weblogic | [logout](#)

Agent:

Visible Rows:

The machine selection display statistical information about the operating system resources currently being used. These would in the Processor, Memory, Swap, File Systems, and running Processes.

Processor | Memory/Swap | Disks | **Process List**

Agent	Name	Pid	User	CPU(%)	Memory (kb)
sox	java	331	pj	0.00	95068
sox	java	411	pj	0.00	19076
sox	Syntmf	182	pj	0.00	12356

4. To turn the Filter off at any time, click Hide Filter.

Starting Statistics Monitoring

To begin monitoring WebLogic Server performance statistics:

1. In the Administration Console, click Performance Monitor.
2. Click the Configuration folder. Select Agent. The Agent Configuration table opens in the right panel.
3. The Default Ruleset is the set of monitoring rules that are configured within the software. The statistics ruleset specifies which server performance statistics to monitor.

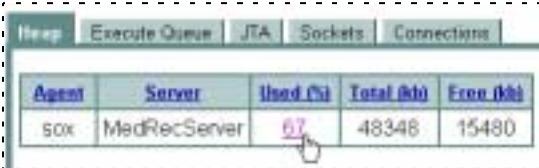
- In the Rules column, select Apply Default Ruleset for each Agent that you want to monitor.
- In the Monitoring Options column, check the Statistics box for each Agent that you want to monitor.
 - Click Apply to start the statistics monitor. See [“Viewing Statistics Monitoring” on page 2-21](#).

Viewing Statistics Monitoring

To view the results of the server statistics monitoring:

- In the Administration Console, click Performance Monitor.
- Click Statistics, then click Servers.
- The Server Statistics view is displayed in the right panel. Click the tabs to view various statistics regarding heap, execute queues, transactions, sockets, and connections.

If a statistic is displayed as a link, usually in a different color and underlined (depending on your browser configuration), you can click the number to open a historical data chart for that statistic.



Agent	Server	Used (Ki)	Total (Ki)	Free (Ki)
soc	MedRecServer	<u>67</u>	48348	15480

The data chart will open in a new window. You can manipulate the chart view to zoom in on segments of data, or to zoom out to see more data. See [“Working with Data Charts” on page 2-22](#).

Note: You can sort the results in the Server Statistics view in ascending or descending order by clicking on any of the column headers. For example, clicking Server will sort the results in alphabetical order based on the WebLogic Server instance name.

You can filter the results to show selected information. See “[Filtering Results](#)” on page 2-19.

Stopping Monitoring

To stop any of the monitoring groups:

1. In the Administration Console, click the Performance Monitor folder.
2. Click the Configuration folder, and click Agent. The Agent Configuration table is displayed in the right column.
3. In the Monitoring Options column, uncheck the box next to the monitoring group that you want to turn off. For example, if you did not want to monitor system resources, uncheck the Resources box.
4. Click Apply in the Monitoring Options column. The changes will take place immediately.

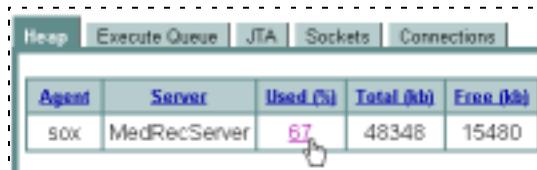
Working with Data Charts

Some pages, such as the Resources—>Machine page and the Statistics—>Servers page, contain links to historical data charts. These charts display statistical values collected by the Performance Monitor over a period of time.

Note: You must use JRE 1.3 or later to chart applets.

You can manipulate the data charts to zoom in on selected portions, or to zoom out to view additional data.

1. Open a data chart by clicking a linked statistic.



Agent	Server	Used (%)	Total (kb)	Free (kb)
sox	MedRecServer	67	48348	15480

2. The data chart opens in a new window. It contains data values collected every five minutes for the previous hour.
 - To zoom out, hold down the shift key and left-click in the chart. You can continue doing this until the entire hour's worth of data is displayed.
 - To zoom in on a portion, hold down the shift key. Left click on the desired start time, hold down the mouse button, and drag to the desired end time as shown below. Release the mouse and shift key.



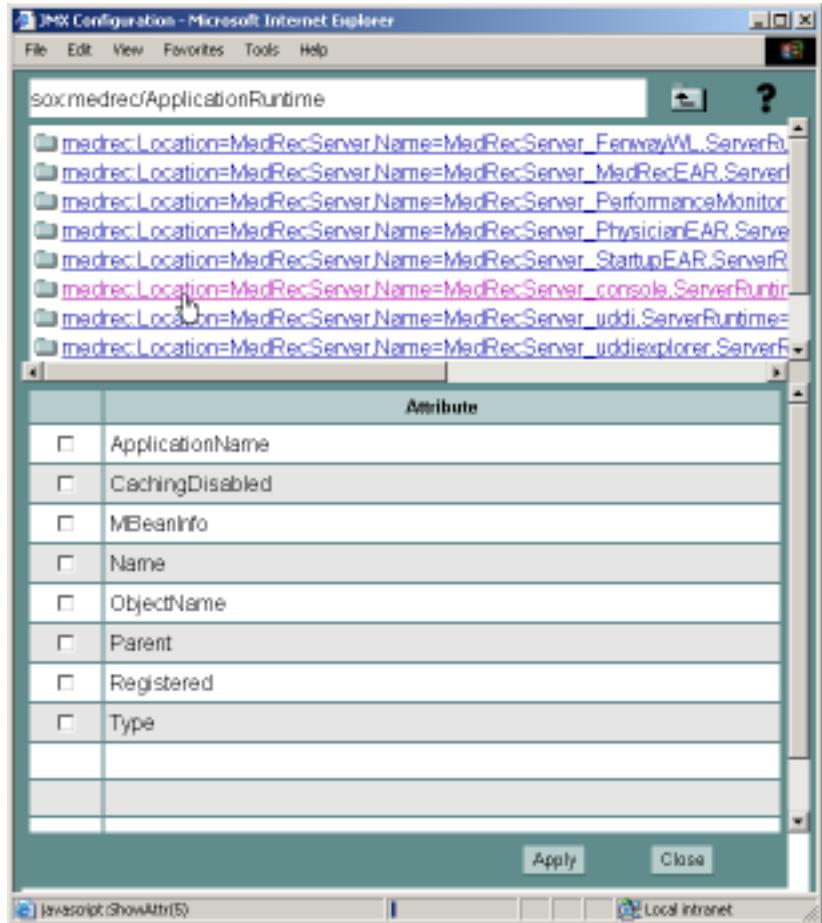
The chart resizes to show the desired time span.

Starting JMX Application Monitoring

To start monitoring JMX MBean attributes within the server:

1. The statistics monitor must be running. See [“Starting Statistics Monitoring” on page 2-20](#).
2. In the Administration Console, click the Performance Monitor.
3. Click the Configuration folder, then click JMX. The JMX Configuration view opens in the right panel.
4. Click the Add MBeans (and attributes)... link.
5. The JMX Configuration view opens in a new window displaying a tree of MBeans organized by type. MBean types are displayed as folders and MBean instances are displayed as leaves. At the top of the view, select the Agent:WebLogic Server instance combination that you want to monitor. All directories that contain MBean objects are displayed.

- Continue clicking through the desired directories until you see a list of MBean objects. Click the desired MBean name. The lower portion of the JMX Configuration view lists all the attributes for the MBean, as shown in the following example.



In the Attribute list, check the box next to each attribute that you want to monitor. Click Apply, and close the JMX Configuration view.

- The JMX Monitor will immediately begin monitoring the attribute values. See [“Viewing JMX Application Monitoring”](#) on page 2-25.

Viewing JMX Application Monitoring

To view the results of JMX Monitoring:

1. In the Administration Console, click the Performance Monitor.
2. Click Statistics, then click JMX.
3. The JMX view opens in the right panel, and displays the following information:
 - Agent: Name of the host machine.
 - Server: Name of the WebLogic server instance.
 - Name: Name of the MBean being monitored.
4. To view the latest attribute values retrieved by the JMX Monitor, click the MBean name in the Name column.
5. The MBean Attribute view opens. Each attribute being monitored is listed, and the most recent value for each attribute is displayed.

JMX Monitor Configuration

Modifying JMX Monitor Configuration

To modify which MBeans and attributes you want to monitor:

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click JMX to open the JMX Configuration view.
3. In the Name field, click the name of the MBean that you want to modify.
4. The MBean Configuration view opens in a new window. To change the MBean being monitored, click the name of the new MBean that you want to monitor. To select an MBean in another directory, use the navigation tools at the top of the window to go to another directory.
5. The attributes for the selected MBean are listed. Check the box next to each attribute that you want to monitor, then click Apply. Close the MBean Configuration view. The changes will take effect immediately.

Deleting MBean Monitoring

To delete MBean monitoring instructions (note that this does not delete the MBean itself):

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click JMX. The JMX Configuration view opens in the right panel.
3. Scroll to the far right of the JMX Configuration view. There is a Delete icon next to the name of each MBean being monitored.

Click the Delete icon next to the desired MBean; this will remove the MBean from monitoring.



JMX Application Monitoring Example

The following example shows how to use JMX MBean monitoring to retrieve performance statistics for JVM heap utilization.

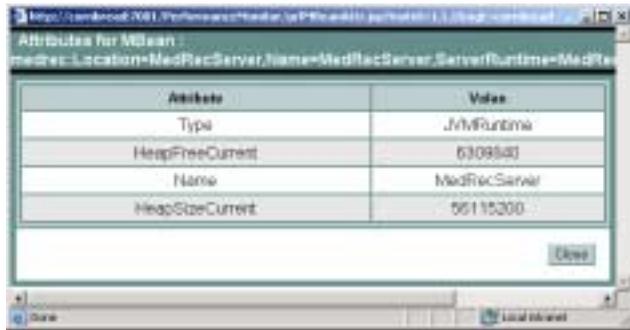
1. The JMX MBean monitor is configured as shown in [“Starting JMX Application Monitoring” on page 2-23](#). The directory path is:
<machine_name>:<WebLogic_Server_domain>/JVM Runtime.



The attributes are selected as shown.

2. The MBean attributes are then viewed as shown in [“Viewing JMX Application Monitoring” on page 2-25](#).

Note the current values for JVM heap utilization. `HeapFreeCurrent` is the size of the heap not in use. `HeapSizeCurrent` is the total size, in bytes, of memory currently allocated to the heap.



Attribute	Value
Type	JVMRuntime
HeapFreeCurrent	5309340
Name	MedRecServer
HeapSizeCurrent	56115300

Any MBean attribute that is exposed can be monitored by the JMX monitor.

Starting Application Monitoring

To start monitoring a deployed application or module:

1. The statistics monitor must be running. See [“Starting Statistics Monitoring”](#) on page 2-20.
2. In the Administration Console, click the Performance Monitor.
3. Click Statistics, then click the folder for the application type or module that you want to monitor. Select Applications, EJB Modules, Web Applications, or Unassociated.
4. The <Application Type> view opens in the right panel. The Agent name, WebLogic Server instance name, and application name are shown for each deployed application of that type. If no applications are displayed, then no applications of this type are currently deployed.
5. To turn monitoring on for an application, check the box in the Enable column, then click Apply.

Stopping Application Monitoring

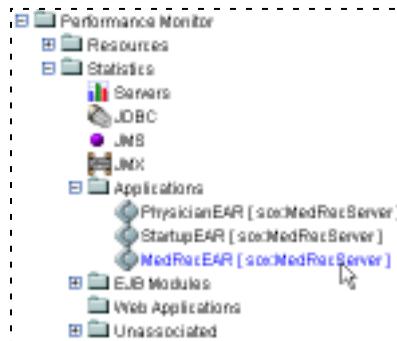
To stop monitoring a deployed application or module:

1. In the Administration Console, click Performance Monitor.
2. Click Statistics, then click the appropriate folder for the application type (Applications, EJB Modules, Web Applications, or Unassociated).
3. In the <Application Type> view, uncheck the box in the Enable column, and click Apply.

Viewing Application Monitoring

To view monitoring of deployed applications or modules:

1. In the Administration Console, click Performance Monitor.
2. Click Statistics, then click the appropriate folder for the application type (Applications, EJB Modules, Web Applications, or Unassociated).
3. The folder will expand to show all applications of that type that are being monitored. To view the monitoring results, click the application name as shown in the following example.



- The <Application type> view opens in the right panel. Click the tabs to view available statistics for the servlets, JSPs, EJBs, and custom classes that make up the application.

If a statistic is displayed as a link, usually in a different color and underlined, you can click the number to open a historical data chart for that statistic.

Servlets JSPs EJBs User Classes						
Agent	Server	Name	Requests per Minute	Response Time (ms)		
				Average	Min	Max
sox	MedRecServer	*jvf	() 0.00	0	0	0
sox	MedRecServer	*jvs	() 0.00	0	0	0

The data chart will open in a new window. You can manipulate the chart view to zoom in on segments of data, or to zoom out to see more data. See [“Working with Data Charts” on page 2-22](#).

- You can run instrumentation on application components, and view method performance statistics for components such as servlets and EJBs. See [“Starting Instrumentation” on page 2-30](#) and [“Viewing Method Statistics” on page 2-33](#).

Note: You can sort the results in the <Application type> view in ascending or descending order by clicking on any of the column headers. For example, clicking Name—for the view pictured above—will sort the results in alphabetical order based on the servlet name.

You can filter the results to show selected information. See [“Filtering Results” on page 2-19](#).

Starting Instrumentation

Prior to turning on instrumentation, you must start the statistics monitor, and enable application monitoring for the desired application:

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

The application must be deployed on the Administration Console. You must also have in-depth knowledge of the application, and how its components are configured.

To turn instrumentation on:

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click Instrumentation.
3. The Instrumentation view is displayed in the right panel, and shows a list of each Agent capable of instrumentation monitoring.

In the Instrumented Classes column, click View... for the desired Agent. This Agent must contain the applications with the associated methods that you want to monitor.

The Instrumentation Configuration view opens in a new window.

4. The top of the Instrumentation Configuration view shows the root directory and sub-directory folders. Click the sub-directory folders to get to your desired file location.

When you select the desired directory, a list of components (such as servlets, JSPs, and EJBs) detected by the Performance Monitor is displayed.

Find the desired component(s). First, check the box next to the component in the Instrument field. This enables instrumentation. Second, check the box in the Tracing field. Tracing is the command used to start collecting method-level statistics.

In the Application field, click the drop-down menu to see a list of enabled applications. These applications are being monitored by the Performance Monitor (See [“Starting Application Monitoring” on page 2-28](#)). Select the application that this component is associated with.

When you’ve turned on instrumentation and tracing for all desired components in this location, and you’ve selected the applications they’re associated with, click Apply to begin monitoring method-level statistics.

The following is an example of a component being instrumented.



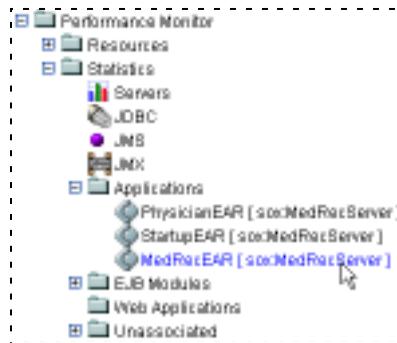
Note: The Instrument command enables a component to be instrumented. Tracing is the on/off command to actually retrieve statistics. If you turn tracing on for a large number of components, it can generate a large amount of data that may be unnecessary. It is recommended that you use tracing only to collect data from components you desire, and during time periods that you're interested in.

5. Restart WebLogic Server.
6. Once you've enabled instrumentation and turned on tracing, you can view the method performance statistics being retrieved by the Performance Monitor. See ["Viewing Method Statistics" on page 2-33](#).

Viewing Method Statistics

To view method-level monitoring retrieved by instrumentation:

1. In the Administration Console, click Performance Monitor.
2. Click Statistics, then click the appropriate folder for the application type (Applications, EJB Modules, Web Applications, or Unassociated).
3. The folder will expand to show all applications of that type that are being monitored. To view the monitoring results, click the application name as shown in the following example.



4. The <Application type> view opens in the right panel. Click the Instrumented Classes tab. Instrumentation must be enabled for components associated with this type of application. See [“Starting Instrumentation”](#) on page 2-30.

Find the component that is running instrumentation. Click the braces next to the name of the component as shown in the following example.

Agent	Server	Name	Response Time (ms)	Iterations	Exceptions
sox	MedRecServer	AddressBean	{}	0	0
sox	MedRecServer	AddressEJB	{}	0	0
sox	MedRecServer	BaseBean	{}	0	0

The Component Methods view opens in a new window, and displays the following information:

- Method Name: Name of the method.
- Rsp Time (ms): Average response time, in ms, for the method to process a request.
- Iteration: Number of times the method has been called.
- Exceptions: Number of exceptions thrown by the method.

The following is an example of viewing method performance monitoring of a component:

Method Name	Rsp Time (ms)	Iteration	Exceptions
setRepositorySelector	0	0	0
getLoggerRepository	0	1	0
getRootLogger	0	1	0
getLogger	0	15	0
getLogger	0	0	0
getLogger	0	0	0
exists	0	0	0
getCurrentLoggers	0	0	0
shutdown	0	0	0
resetConfiguration	0	0	0

5. You can view historical performance statistics for methods displayed in the Component Methods view. Click the method name or any of its statistics to view data charts for the method's response time and iteration counts. You can manipulate the data chart as shown in [“Working with Data Charts”](#) on page 2-22.

Stopping Tracing

To stop tracing:

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click Instrumentation.

3. The Instrumentation view is displayed in the right panel, and shows a list of each Agent capable of instrumentation.

In the Instrumented Classes column, click the View... button for the desired Agent. This Agent must contain the applications with the associated methods that you want to monitor.

The Instrumentation Configuration view opens in a new window.

4. The top of the Instrumentation Configuration view shows the root directory and sub-directory folders. Click the sub-directory folders to get to your desired file location.

Find the desired component(s). Uncheck the box in the Tracing field. Click Apply.

Note: This will only temporarily disable tracing. If the WebLogic server is restarted, tracing will start again. To permanently disable tracing for a component, uncheck the Instrument box as well. The next time the server is restarted, instrumentation will not be enabled for that object, and as a result tracing will be disabled as well.

Stopping All Tracing Using Global Disable

To stop all tracing at once using the Global Disable button:

1. In the Administration Console, click Performance Monitor.
2. Click Configuration, then click Instrumentation.
3. Click the Global Disable button to turn off all tracing for that Agent. Note that this is only temporary, and tracing will restart if the server is restarted.

Instrumentation Example

The following is an example of using instrumentation to monitor method performance.

1. Instrumentation is configured as shown in “Starting Instrumentation” on page 2-30. In this example, methods for the PatientEJB WebLogic Server entity will be monitored.



2. Once instrumentation is configured, WebLogic Server is restarted.
3. Results can be viewed as shown in “Viewing Method Statistics” on page 2-33. In this example, the Unassociated node in the left panel is selected. The right panel displays the Instrumented Classes.

Instrumented Classes						
Agent	Server	Name	Response Time (ms)	Iterations	Exceptions	
com.bea.medrec	MedRec-Server	com.bea.medrec.entities.PatientEJB { }	4	15	0	

- To view method performance statistics, click { } next to the class name.

Agent	Server	Name	Response Time (ms)
combread	MedRecServer	com.bea.medrec.entities.PatientEJB	4

- The **Methods for: <Entity_name>** view opens. All methods associated with this entity are displayed. For example, you can see that the method `getPatient` has been called once, and the average response time for it was 42 milliseconds.

Method Name	Resp Time (ms)	Iterations	Exceptions
<code>getPatient</code>	42	1	0
<code>getEjbCreate</code>	22	1	0
<code>getFirstName</code>	0	0	0
<code>getMiddleName</code>	0	0	0
<code>getLastName</code>	0	0	0
<code>getJob</code>	0	0	0
<code>getGender</code>	0	0	0
<code>getSex</code>	0	0	0
<code>getPhone</code>	0	0	0
<code>getEmail</code>	0	0	0
<code>getAddress</code>	0	0	0
<code>setFirstName</code>	0	0	0
<code>setMiddleName</code>	0	0	0
<code>setLastName</code>	0	0	0
<code>setJob</code>	0	0	0
<code>setGender</code>	0	0	0
<code>setSex</code>	0	0	0
<code>setPhone</code>	0	0	0
<code>setEmail</code>	0	0	0
<code>setAddress</code>	0	0	0
<code>total</code>	0	0	0



3 Troubleshooting the Performance Monitor

Table 3-1 Troubleshooting the Performance Monitor

Problem	Solution
The Performance Monitor folder does not appear in NavApplet.	Make sure the software was installed in the appropriate WebLogic home directory. The WebLogic Server will automatically look for the console extension in the \$WL_HOME/common/perf directory. The console extension is called PerformanceMonitor.war.
No data appears when clicking on any of the resources links within the NavApplet.	Most likely the agent process for the monitored server is not running. The statistical information for these views is actually stored within the Performance Monitor external agent.
No data appears when clicking on any link within the Performance Monitor Statistics section.	The statistical information for this section relies on the deployment of the Performance Monitor web application as well as the agent process (see #2). This web application can be deployed by clicking on the Performance Monitor link in the NavApplet and selecting the Deployments tab.
My applications do not show up under the Statistics folder of the Performance Monitor.	The Performance Monitor, by default, does not monitor any application without explicit direction from the user. To monitor a specific application or EJB module, select the application type folder within the NavApplet. When the list of deployed applications appear, select any number of applications to monitor.
I do not have any pre-configured Thresholds, Logs or Network Service tests.	In the unlikely event that the rules do not appear, the factory default rules can be dynamically created. This option can be found under the Agent section of the Configuration folder. The “Apply Default Ruleset” will create a set of “out-of-the-box” factory default rules. This feature can also be used to reset any previous configured rules.

Table 3-1 Troubleshooting the Performance Monitor

The WebLogic console window displays hostname verification errors.	In certain circumstances, hostname verification needs to be disabled. Please refer to the WebLogic Server users guide for further instructions.
I would like to have the agent restart when I restart the server.	Please follow the directions entitled “Starting the Agent”, found in this document.
The agent failing upon startup, or does not seem to be operational.	The Agent has a command line switch to output a series of messages that are useful in debugging most situations. To enable debug start the agent with the -debug -console switch.

Table 3-1 Troubleshooting the Performance Monitor

I seem to be experiencing certificate conflicts.

You can manually regenerate the Performance Monitor Agent certificate by issuing the following command:

```
java -cp
$WL_HOME/common/perf/java/dirig.jar;$WL_HOME/server/lib
/weblogic.jar com.dirig.certs.ConvertCerts $WL_HOME
<java> true
```

where \$WL_HOME is WebLogic home and <java> is Java home. The “true” parameter controls the output of debug information during the generation process. This switch is vital to figuring out where the problem could be. For example:

```
java -cp
common\perf\java\dirig.jar;server\lib\weblogic.jar
com.dirig.certs.ConvertCerts \bea81sp1\weblogic81
\bea81sp1\jdk141_03 true
```

The output is as follows:

```
DEMO_IDENTITY_AND_DEMO_TRUST
trustPassword : null
getKeyStore
\bea81sp1\weblogic81\server\lib\DemoIdentity.jks jks
DemoIdentityKey
StorePassPhrase
convertPrivateKey DemoIdentity DemoIdentityPassPhrase
\bea81sp1\weblogic81\commo
n\perf\lib\key.pem
convertCertChain DemoIdentity
\bea81sp1\weblogic81\common\perf\lib\cert.pem
getKeyStore
\bea81sp1\weblogic81\server\lib\DemoTrust.jks jks
DemoTrustKeyStoreP
assPhrase
converttrustcas
\bea81sp1\weblogic81\common\perf\lib\cas.pem
getKeyStore \bea81sp1\weblogic81\server\lib\cacerts jks
null
converttrustcas
\bea81sp1\weblogic81\common\perf\lib\cas.pem
```

Note: This output may contain sensitive information.



Performance Monitor --> Resources --> Machine --> Process List

Tasks

Overview

The Resources—>Machine—>Process List tab displays statistics for all processes currently running on a server. These statistics are retrieved by the resource monitor:

- Agent: Name of the server being monitored.
- Name: Process name.
- PID: Process Identification number assigned to the process by the operating system.
- User: Name of the user currently logged on to the server.
- CPU (%): Amount of CPU capacity being consumed by the process.
- Memory (Kb): Amount of memory (RAM) being consumed by the process.

These statistics provide insight into the operating performance of your servers. They can serve as an initial indicator of potential problems. For example, if a process is consuming an abnormally large amount of CPU capacity, you may need to restart the process, or troubleshoot the program's code.

Tasks

[“Starting Machine System Resource Monitoring” on page 2-14](#)

[“Viewing Machine System Resource Monitoring” on page 2-14](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)



Performance Monitor --> Configuration --> Instrumentation

Tasks

Overview

The Configuration—>Instrumentation page displays all servers with instrumentation capabilities. Use this page to instrument selected components, and to turn tracing on to retrieve statistics for methods operating within the components. You can also turn tracing off for all components using the Global Disable button.

Instrumentation is a tool you can use to increase the capabilities of statistics monitoring. The statistics monitor retrieves performance statistics for application and module components; instrumentation allows you to retrieve performance statistics for each method operating within objects such as servlets or EJBs.

Before instrumentation can be configured, the statistics monitor must be turned on, and application must be enabled for the desired application or module.

There are two parts to instrumentation. First, instrumentation needs to be enabled for the selected component(s). The Performance Monitor cannot execute tracing on a component until instrumentation is enabled. Second, tracing needs to be turned on for a component that is instrumented. Tracing is the actual action of monitoring and retrieving statistics from methods.

Instrumentation is an advanced feature, and requires the user to have in-depth knowledge of how an application and its components are configured. You must know which components are operating within an application prior to configuring instrumentation for any WebLogic components.

The instrumentation configuration allows you to turn tracing on or off for the components you want to monitor. This can greatly minimize system resource consumption, and reduce unnecessary data generation. It is recommended that you use tracing only for desired components, and during desired time periods.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Tracing” on page 2-34](#)

Performance Monitor --> Configuration --> JMX

[Tasks](#) [Related Topics](#)

Overview

The Configuration—>JMX page displays all MBean objects currently being monitored by the Performance Monitor. Use this page to configure MBean objects and their attributes for monitoring, and also to delete monitoring rules for MBean objects.

A JMX application is a program based on the Java Management Extensions (JMX) standard. MBeans objects operate within the JMX application to perform tasks. Attributes are the various characteristics associated with an MBean.

You can configure the Performance Monitor to monitor any MBean object attributes operating within the server. You can view the most recent attribute values retrieved by the statistic monitor.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting JMX Application Monitoring” on page 2-23](#)

[“Viewing JMX Application Monitoring” on page 2-25](#)

[“JMX Monitor Configuration” on page 2-25](#)

[“Modifying JMX Monitor Configuration” on page 2-25](#)

[“Deleting MBean Monitoring” on page 2-26](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[“Overview of WebLogic JMX Services”](#)

Performance Monitor --> Configuration --> Log

[Tasks](#) [Related Topics](#)

Overview

The Configuration—>Logs page displays the log monitoring currently configured for the Performance Monitor. Use this page to view and modify configuration details for the log monitor, or to delete the log monitoring rules.

The log monitor is used to scan log file entries for specific text patterns. These entries are compared to regular expression patterns specified in the log monitor. If any part of an entry matches the pattern, an alert is generated.

This is useful for detecting security or performance errors occurring within the server, and subsequently logged by the server. For example, the log monitor is configured to monitor log file entries for the occurrence of the words “HTTP” and “error”. The presence of these terms could indicate page load failures, and the administrator can be quickly notified by an alert.

Tasks

[“Starting Log File Monitoring” on page 2-8](#)

[“Viewing Log File Monitoring” on page 2-9](#)

[“Log Monitor Configuration” on page 2-10](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[Overview of WebLogic Server Log Messages and Log Files](#)

Performance Monitor --> Configuration --> Network Service Test

[Tasks](#) [Related Topics](#)

Overview

The Resources—>Network Service Test page displays the network service tests currently configured for the Performance Monitor. Use this page to view and modify configuration details for the network service tests, or to delete monitoring rules for network service tests.

The Network Service Test is used to check HTTP performance by periodically attempting to download a Web page from a remote server. The time required to connect and download the page is recorded by the Performance Monitor. This test is useful for identifying problems with HTTP transmissions within the network.

If the page fails to download, an alert is generated. A threshold is configured to set operating parameters for the response time to download the page. If the response time exceeds the threshold parameters, an alert will be generated.

Tasks

[“Starting Network Service Tests” on page 2-12](#)

[“Viewing Network Service Tests” on page 2-12](#)

[“Network Service Test Configuration” on page 2-13](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[Configuring the HTTP Protocol](#)

Performance Monitor --> Configuration --> Notifications

Tasks

Overview

The Configuration—>Notification page displays the current information for notification instructions. Use this page to turn e-mail notification on when an alert occurs, and to configure the recipient and sender e-mail addresses for notification.

The Performance Monitor uses alerts to declare when it detects a problem. Enabling notifications will instruct the Performance Monitor to send an e-mail message immediately upon generation of an alert.

This feature is useful for active notification of events. An administrator needs to periodically review the Alerts view in the Performance Monitor to see existing alerts; notifications provides a way to more actively and immediately alert an administrator of a problem.

Tasks

[“Viewing Alerts” on page 2-7](#)

[“Turning On E-mail Notification” on page 2-7](#)



Performance Monitor --> Configuration --> Thresholds

Tasks

Overview

The Configuration—>Threshold page displays the thresholds currently configured for the Performance Monitor. Use this page to view and modify threshold configuration, or to delete threshold monitoring rules.

A threshold is a set of rules that create operating parameters for an attribute. If the attribute's statistical value falls outside the parameters, an alert is generated.

For example, one of the thresholds monitors the percentage of memory in used. First, the threshold measures the percentage of memory in use every 30 seconds for a period of three minutes to establish a normal operating baseline.

Once the baseline is established, the threshold begins comparing actual values to the parameters established by the threshold. In the memory example, the threshold states that the percentage of memory used must not rise more than 25% above the baseline value. If the baseline is 40%, then a value greater than 50% ($40\% + (40\% \cdot 0.25)$) will generate an alert.

Thresholds are extremely useful for notifying you of an adverse condition before failure occurs. Continuing with the memory example, the threshold will alert you of abnormal memory consumption before it leads to a major malfunction. You can then diagnose and resolve the problem before it affects system performance.

The default thresholds that come with the Performance Monitor are configured to monitor these attributes:

- CPU Usage
- Memory Usage
- Network Service Test Response Times
- JVM Heap Utilization

Tasks

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

Performance Monitor --> Resources --> Alerts

Tasks

Overview

The Resources—>Alerts page displays alerts that are generated by the Performance Monitor when it detects a problem. All alerts present on the server can be viewed on this page.

Alerts are the main vehicle the Performance Monitor uses to declare when an adverse condition exists. For example, if a log monitor finds a pattern match in a log file entry (which could indicate a problem within an application), the Performance Monitor generates an alert.

The alert includes information such as what time the alert occurred, a brief text message that details the alert, and a severity level. Each alert is assigned one of three severity levels based on how it may affect system performance:

- **Low (Yellow color code):** Alerts with this severity usually result from problems that are informational in nature, and do not inhibit system performance. For example, the log monitor may detect someone other than an administrator attempting to log onto the WebLogic console. The system is not degraded, but you now are aware of unauthorized users attempting to log on to the system.
- **Medium (Orange color code):** An alert with a medium severity rating is generated by a problem that could affect system performance. For example, a threshold may be monitoring memory usage. If the memory usage rises above a certain point, it could cause the system to slow down considerably.
- **High (Red color code):** High alerts result from problems that are extremely detrimental to system performance, and need to be addressed immediately. An example may be a failed network service test between the Performance Monitor and a critical web server. An unavailable server would need to be recognized and resolved as soon as possible.

You can configure the Performance Monitor to send you an e-mail whenever an alert occurs. This provides immediate notification when an event happens, and alleviates the need to constantly check the Alerts view.

Tasks

[“Viewing Alerts” on page 2-7](#)

[“Turning On E-mail Notification” on page 2-7](#)

Performance Monitor --> Resources --> Logs

[Tasks](#) [Related Topics](#)

Overview

The Resources—>Logs page displays the results of monitoring the WebLogic Server log file.

The log monitor is used to scan log file entries for specific text patterns. These entries are compared to regular expression patterns specified in the log monitor. If any part of an entry matches the pattern, an alert is generated.

This is useful for detecting security or performance errors occurring within the server, and subsequently logged by the server. For example, the log monitor is configured to monitor log file entries for the occurrence of the words “HTTP” and “error”. The presence of these terms could indicate page load failures, and the administrator can be quickly notified by an alert.

This page displays information on the log file being monitored, and prominently displays the number of pattern matches that have occurred. It also includes a link to view log file entries.

Tasks

[“Starting Log File Monitoring” on page 2-8](#)

[“Viewing Log File Monitoring” on page 2-9](#)

[“Log Monitor Configuration” on page 2-10](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

[“Viewing Alerts” on page 2-7](#)

[“Turning On E-mail Notification” on page 2-7](#)

Related Topics

[Overview of WebLogic Server Log Messages and Log Files](#)

Performance Monitor --> Resources --> Machine --> Disks --> File Systems

Tasks

Overview

The Resources—>Machine—>Disks—>File Systems tab displays file system statistics for each server. A file system denotes a portion of a disk designated as a single file system, and can be an entire hard disk drive or a virtual partition of one.

These statistics are retrieved by the resource monitor:

- Agent: Name of the server being monitored.
- Name: Name of the file system instance.
- Used (%): Total percentage of the file system currently in use.
- Size (Kb): Total size of the file system, in Kb.

These statistics provide insight into the operating performance of your servers. They can serve as an initial indicator of potential problems. For example, if the percentage of a file system consumed is high (as an example, over 90%), you may need to add disk space or archive unnecessary files.

Tasks

[“Starting Machine System Resource Monitoring” on page 2-14](#)

[“Viewing Machine System Resource Monitoring” on page 2-14](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Performance Monitor --> Resources --> Machine --> Disks --> Physical Disks

Tasks

Overview

The Resources—>Machine—>Disks—>Physical Disks tab displays physical hard disk drive statistics for each server. The physical disk refers to an actual physical hard disk drive in the machine.

These statistics are retrieved by the resource monitor:

- Agent: Name of the server being monitored.
- Physical Disk: Name of the physical hard disk drive.
- Percent Busy: Total percentage of the physical disk drive currently in use.

These statistics provide insight into the operating performance of your servers. They can serve as an initial indicator of potential problems. For example, if the percentage of a physical disk consumed is high (as an example, over 90%), you may need to add disk space or archive unnecessary files.

Tasks

[“Starting Machine System Resource Monitoring” on page 2-14](#)

[“Viewing Machine System Resource Monitoring” on page 2-14](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)



Performance Monitor --> Resources --> Machine --> Memory/Swap

Tasks

Overview

The Resources—>Machine—>Memory/Swap tab displays memory and swap performance statistics for each server. These statistics are retrieved by the resource monitor:

- Agent: Name of the server being monitored.
- Memory (%): Total percentage of memory (RAM) capacity currently in use.
- Swap (%): Total percentage of swap capacity in use.

These statistics provide insight into the operating performance of your servers. They can serve as an initial indicator of potential problems. For example, if the percentage of memory consumed is high (as an example, over 90%) for a long period of time, it may indicate a problem with an application. You can then use other tools in the Performance Monitor to troubleshoot the problem before a failure occurs.

Tasks

[“Starting Machine System Resource Monitoring” on page 2-14](#)

[“Viewing Machine System Resource Monitoring” on page 2-14](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Performance Monitor --> Configuration --> Agent

Tasks

Overview

The Configuration—>Agent page displays monitoring options that can be turned on or off for each server that has the Performance Monitor installed. Use this page to turn monitoring on or off for these groups:

- **Resources:** Monitors system resources such as CPU, files system, and memory utilization.
- **Log Monitoring:** Monitors log files for specific entries that may indicate problems occurring in an application.
- **Network Service Test:** Performs network tests to monitor availability and performance of a remote server.
- **Statistics:** Monitors various statistics related to WebLogic Server instances, their resources, and their deployed applications.

Tasks

[“Starting Log File Monitoring” on page 2-8](#)

[“Starting Network Service Tests” on page 2-12](#)

[“Starting Machine System Resource Monitoring” on page 2-14](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Starting Statistics Monitoring” on page 2-20](#)

[“Stopping Monitoring” on page 2-22](#)



Performance Monitor --> Resources --> Machine --> Processor

Tasks

Overview

The Resources—>Machine—>Processor tab displays processor performance statistics for each server. These statistics are retrieved by the resource monitor, and are shown for each processor (CPU) instance:

- Agent: Name of the machine being monitored.
- Processor: Physical instance number of the CPU.
- Used (%): Total percentage of CPU capacity currently in use.
- Idle (%): Total percentage of CPU capacity not in use.
- User (%): Percentage of CPU consumed by the current user.
- Kernel (%): Percentage of CPU consumed by the kernel.
- Wait (%): Percentage of CPU consumed by wait states.

These statistics provide insight into server operating performance. They can serve as an initial indicator of potential problems. For example, if the percentage of CPU consumed is high (as an example, over 90%) for a long period of time, it may indicate a problem with an application. You can then use other tools in the Performance Monitor to troubleshoot the condition before a failure occurs.

One of the supplied thresholds monitors CPU usage. If the total percentage of CPU in use exceeds the parameters established by the threshold, an alert will be generated.

Tasks

[“Starting Machine System Resource Monitoring” on page 2-14](#)

[“Viewing Machine System Resource Monitoring” on page 2-14](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Performance Monitor --> Resources --> Network Service Test

[Tasks](#) [Related Topics](#)

Overview

The Resources—>Network Service Test page displays the results of HTTP protocol testing.

The Network Service Test is used to check HTTP performance by periodically attempting to download a Web page from a remote server. The time required to connect and download the page is recorded by the Performance Monitor. This test is useful for identifying problems with HTTP transmissions within the network.

If the page fails to download, an alert is generated. A threshold is configured to set operating parameters for the response time to download the page. If the response time exceeds the threshold parameters, an alert will be generated.

Tasks

[“Starting Network Service Tests” on page 2-12](#)

[“Viewing Network Service Tests” on page 2-12](#)

[“Network Service Test Configuration” on page 2-13](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

[“Viewing Alerts” on page 2-7](#)

[“Turning On E-mail Notification” on page 2-7](#)

Related Topics

[Configuring the HTTP Protocol](#)

Performance Monitor --> Resources --> Thresholds

Tasks

Overview

The Resources—>Threshold page displays the results of threshold monitoring. A threshold is a set of rules that create operating parameters for an attribute. If the attribute's statistical value falls outside the parameters, an alert is generated.

For example, one of the thresholds monitors the percentage of memory in used. First, the threshold measures the percentage of memory in use every 30 seconds for a period of three minutes to establish a normal operating baseline.

Once the baseline is established, the threshold begins comparing actual values to the parameters established by the threshold. In the memory example, the threshold states that the percentage of memory used must not rise more than 25% above the baseline value. If the baseline is 40%, then a value greater than 50% ($40\% + (40\% \cdot .25)$) will generate an alert.

Thresholds are extremely useful for notifying you of an adverse condition before failure occurs. Continuing with the memory example, the threshold will alert you of abnormal memory consumption before it leads to a major malfunction. You can then diagnose and resolve the problem before it affects system performance.

The default thresholds are configured to monitor these attributes:

- CPU Usage
- Memory Usage
- Network Service Test Response Times
- JVM Heap Utilization

Tasks

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

[“Stopping Monitoring” on page 2-22](#)

Performance Monitor --> Statistics --> Applications

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Applications page displays all deployed applications that can be monitored. An application is a J2EE-based application contained in an Enterprise archive file (EAR) or EAR directory.

This page is used to enable statistics monitoring for selected applications. Monitoring applications will yield statistics for the number of requests made on components (such as servlets, EJBs, and JSPs) within the application, and response times for the component to process requests.

If no applications are listed on this page, then no applications are currently deployed.

You can also configure and run instrumentation on application components. This provides statistics on the number of requests and response times for each method operating within application components.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)

Related Topics

[Deploying New Applications and Modules](#)

Performance Monitor --> Statistics --> EJB Modules

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>EJB Modules page displays all deployed standalone EJB modules (standalone EJB modules are not deployed as part of another application archive) that can be monitored. An Enterprise JavaBean (EJB) module represents one or more EJBs contained in an EJB Java archive (JAR) file or directory.

This page is used to enable statistics monitoring for selected EJB modules. Monitoring EJB modules will yield statistics for the number of requests made on module components, and response times for the component to process requests.

If no standalone EJB modules are listed on this page, then no EJB modules are currently deployed.

You can also configure and run instrumentation on EJB module components. This provides statistics on the number of requests and response times for each method operating within EJB module components.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)

Related Topics

The Administration Console provide additional tabs for monitoring EJBs. See:

- [“EJB Runtime Monitoring”](#)
- [“Tuning In Response to Monitoring Statistics”](#)

Performance Monitor --> Statistics --> JDBC

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>JDBC tab displays JDBC connection pool statistics for a server. A JDBC connection pool contains a group of JDBC connections that are created when the connection pool is registered, usually when starting up WebLogic Server. Your application borrows a connection from the connection pool, uses it, then returns it to the connection pool by closing it.

These connection pool statistics are retrieved by the statistics monitor:

- **Agent:** Name of the host machine.
- **Server:** Name of the WebLogic server instance.
- **Name:** Name of the JDBC Connection Pool.
- **Size:** Current size of the connection pool; shows how many connections are currently assigned to the pool.
- **Used (%):** Percentage of connection pool in use.
- **Number Waiting:** Number of transactions waiting for a connection.
- **Wait Time:** Average time, in ms, for a transaction to wait for a connection.

These statistics can provide insight into server performance. If the percentage of the pool in use is high, or there is a large number of waiting transactions, this may indicate a need to allocate more resources to the server.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[“How JDBC Connection Pools Enhance Performance”](#)

[“Performance Tuning Your JDBC Application”](#)

Performance Monitor --> Statistics --> JMS --> Destinations

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>JMS—>Destinations tab displays statistics regarding the destinations configured for JMS servers. A destination can be a queue or a topic. Destination statistics show the performance and availability of each destination configured for the JMS server.

These JMS Destination statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance.
- Name: Name of the JMS server.
- Consumers: Number of queues or topics available to process messages for the JMS server.
- Messages #: Total number of messages (pending and received) in the destination.
- Messages Pending: Number of messages pending in a destination.
- Messages Received: Number of messages received by the destination.
- Bytes Pending: Total size, in bytes, of messages pending.
- Bytes Received: Total size, in bytes, of messages received.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[Tuning Distributed Destinations](#)

Performance Monitor --> Statistics --> JMS --> Durables

Tasks

Overview

The Statistics—>JMS—>Durables tab displays JMS Durables statistics. Durable subscribers can be configured for JMS servers to store messages until the messages can be delivered to the subscriber. These statistics show the performance and availability of durable subscribers for the JMS server.

These JMS durable subscriber statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Name: Name of the Durable Subscriber.
- Consumers: Number of queues or topics available to process messages for the JMS server.
- Active: Number of active JMS destinations running on the JMS.
- Messages #: Total number of messages (pending and received) for durable subscribers.
- Messages Pending: Number of messages pending for the subscriber.
- Messages Received: Number of messages received by the subscriber.
- Bytes Pending: Total size, in bytes, of messages pending.
- Bytes Received: Total size, in bytes, of messages received.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Performance Monitor --> Statistics --> JMS --> Servers

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>JMS—>Servers tab displays JMS server statistics. JMS servers are configured to manage connections and message requests for JMS clients. JMS server statistics show performance and availability of all JMS servers configured for each WebLogic Server instance.

These JMS server statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server: Name of the WebLogic server instance.
- Name: Name of the JMS server.
- Destinations: Number of destinations (such as queues or topics) configured for the JMS server.
- Messages #: Total number of messages (pending and received) in the JMS server.
- Messages Pending: Number of messages pending in a queue.
- Messages Received: Number of messages received by the JMS server.
- Bytes Pending: Total size, in bytes, of messages pending.
- Bytes Received: Total size, in bytes, of messages received.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[JMS: Tuning](#)

Performance Monitor --> Statistics --> JMX

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>JMX page displays a list of JMX-based MBeans that are being monitored by the Performance Monitor.

A JMX application is a program based on the Java Management Extensions (JMX) standard. MBean objects operate within the JMX application to perform the work required for the application. Attributes are the various characteristics associated with an MBean.

You can configure the Performance Monitor to monitor any MBean attributes operating within WebLogic Server instances. You can view MBean attribute values as they are retrieved by the statistics monitor.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting JMX Application Monitoring” on page 2-23.](#)

[“Viewing JMX Application Monitoring” on page 2-25](#)

[“JMX Monitor Configuration” on page 2-25](#)

Related Topics

[“Overview of WebLogic JMX Services”](#)



Performance Monitor --> Statistics --> Servers --> Connections

Tasks

Overview

The Statistics—>Servers—>Connections tab displays connection statistics for a server. Connections pertain to transmissions from remote agents to a WebLogic Server instance.

These connection statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server In: WebLogic server instance running in its own JVM.
- Remote Address: IP address for the remote agent connecting to the JVM.
- Local: Administrator server name.
- Remote: Name of the remote server sending the message.
- Messages: Number of messages in the queue.
- Bytes: Total size of the messages, in bytes.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)



Performance Monitor --> Statistics --> Servers --> Execute Queue

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Servers—>Execute Queue tab displays execute queue and thread statistics. Execute Queues contain threads that perform tasks requested by WebLogic Server instances.

These queue statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance being monitored.
- Name: Name of the execute queue.
- Queue Length: Number of requests currently in the queue.
- Queue Throughput: Total number of requests that have been processed through the queue.
- Thread Count: Total number of threads assigned to the queue.
- Idle Threads: Number of threads not in use.

Queue statistics provide insight into the server's ability to process work. If queue lengths are long, or most of the threads are constantly being used, you may need to add queues or threads to increase capacity.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[Setting Thread Count](#)

Performance Monitor --> Statistics --> Servers --> Heap

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Servers—>Heap tab displays heap statistics for each server. Heap represents the memory allocated to the server. A server represents a single instance of a WebLogic Server that runs in its own Java Virtual Machine (JVM).

These heap statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance being monitored.
- Used (%): Total percentage of heap currently in use.
- Total (Kb): Total heap capacity, in Kb.
- Free (Kb): Amount of heap not in use, in Kb.

Heap statistics provide insight into the operating performance of your servers, and can warn you of adverse conditions that may affect the server’s functions.

One of the default thresholds is configured to monitor JVM heap performance. If the percentage of heap in use exceeds a specified amount, an alert is generated by the Performance Monitor.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Starting Threshold Monitoring” on page 2-15](#)

[“Viewing Threshold Monitoring” on page 2-16](#)

[“Threshold Configuration” on page 2-17](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[JVM Heap Size and Garbage Collection](#)

[Specifying Heap Size Values](#)

Performance Monitor --> Statistics --> Servers --> JTA

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Servers—>JTA tab displays Java Transaction API (JTA) transaction statistics for each server. Java Transaction API (JTA) allows you to configure transaction processing for your environment. JTA in this instance refers to performance statistics related to transaction processing.

These transaction statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance being monitored.
- Count: Total number of transactions called.
- Commits: Number of transactions successfully processed.
- Rollbacks: Number of failed transactions.
- Begins: Number of transactions initiated and in progress.
- Resumes: Number of transactions that have resumed processing after being suspended.
- Suspends: Number of transactions that have been stopped.

These statistics can provide insight into how effectively a server is processing transactions. A significant number of failed or suspended transactions could indicate a problematic server, or the need to allocate more resources to the server.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Working with Data Charts” on page 2-22](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[JTA](#)

Performance Monitor --> Statistics --> Servers --> Sockets

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Servers—>Sockets tab displays available sockets for each server. Sockets are opened for the transmission of messages from a WebLogic Server instance to remote agents.

These socket statistics are retrieved by the statistics monitor:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance being monitored.
- Remote Address: IP address for the open socket.
- Protocol: Protocol type for the open socket.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Viewing Statistics Monitoring” on page 2-21](#)

[“Filtering Results” on page 2-19](#)

[“Stopping Monitoring” on page 2-22](#)

Related Topics

[Allocating Threads to Act as Socket Readers](#)



Performance Monitor --> Statistics --> Unassociated

Tasks

Overview

The Statistics—>Unassociated page displays monitoring of an unassociated application used for instrumentation. This is a custom application included with the Performance Monitor, and consists of custom user classes. It is used to provide instrumentation for methods running within applications and modules. It is not deployed like other applications; it runs automatically within the Performance Monitor.

This page is used to enable statistics monitoring for the unassociated program.

You can configure and run instrumentation on application components. This provides statistics on the number of requests and response times for each method operating within application components.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)



Performance Monitor --> Statistics --> Web Applications

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Web Applications page displays all deployed Web application modules that can be monitored. A Web application module is a collection of resources such as JavaServer Pages (JSPs), servlets and HTML files contained in a Web application archive file (WAR) or WAR directory.

This page is used to enable statistics monitoring for selected Web application modules. Monitoring Web application modules will yield statistics for the number of requests made on module components, and response times for the component to process requests.

If no Web application modules are listed on this page, then no Web application modules are currently deployed.

You can also configure and run instrumentation on Web application module components. This provides statistics on the number of requests and response times for each method operating within module components.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)

Related Topics

[Deploying New Applications and Modules](#)

Performance Monitor --> Statistics --> Application Module --> EJBs

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Application Module—>EJBs tab displays performance statistics for active Enterprise JavaBeans (EJBs) associated with the selected application. The following information is displayed:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance.
- Name: EJB name.
- Requests Per Minute: Average number of calls made per minute on the EJB.
- Response Time (ms): The average, minimum, and maximum response times, in ms, for the EJB to process a request.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)

Related Topics

The Administration Console provide additional tabs for monitoring EJBs. See:

- [“EJB Runtime Monitoring”](#)
- [“Tuning In Response to Monitoring Statistics”](#)

Performance Monitor --> Statistics --> Application Module --> JSPs

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Application Module—>JSPs tab displays performance statistics for active JavaServer Pages (JSPs) associated with the selected application. The following information is displayed:

- Agent: Name of the host machine.
- Server: Name of the WebLogic server instance.
- Name: JSP name.
- Requests Per Minute: Average number of calls made per minute on the JSP.
- Response Time (ms): The average, minimum, and maximum response times, in milliseconds (ms), for the JSP to successfully process a request.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)

Related Topics

[“Overview of Java Server Pages \(JSP\)”](#)

Performance Monitor --> Statistics --> Application Module --> Servlets

[Tasks](#) [Related Topics](#)

Overview

The Statistics—>Application Module—>Servlets tab displays performance statistics for active servlets running within the selected application. The following information is displayed:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance.
- Name: Servlet name.
- Requests Per Minute: Average number of calls made per minute on the servlet.
- Response Time (ms): The average, minimum, and maximum response times, in milliseconds (ms), for the servlet to process a request.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Working with Data Charts” on page 2-22](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)

Related Topics

[“Overview of HTTP Servlets”](#)

Performance Monitor --> Statistics --> Application Module --> Instrumented Classes

Tasks

Overview

The Statistics—>Application Module—>Instrumented Classes tab displays performance statistics for active instrumented classes associated with the selected application. An instrumented class is any custom class that is not classified as a servlet, JSP, or EJB.

The following information is displayed:

- Agent: Name of the host machine.
- Server: Name of the WebLogic Server instance.
- Name: User Class name.
- { }: These braces link to a view showing all methods associated with the component, and method performance statistics if instrumentation is turned on.
- Requests Per Minute: Average number of calls made per minute on the class.
- Response Time (ms): The average, minimum, and maximum response times, in ms, for the class to process a request.

Tasks

[“Starting Statistics Monitoring” on page 2-20](#)

[“Starting Application Monitoring” on page 2-28](#)

[“Viewing Application Monitoring” on page 2-29](#)

[“Starting Instrumentation” on page 2-30](#)

[“Viewing Method Statistics” on page 2-33](#)

[“Stopping Application Monitoring” on page 2-29](#)



