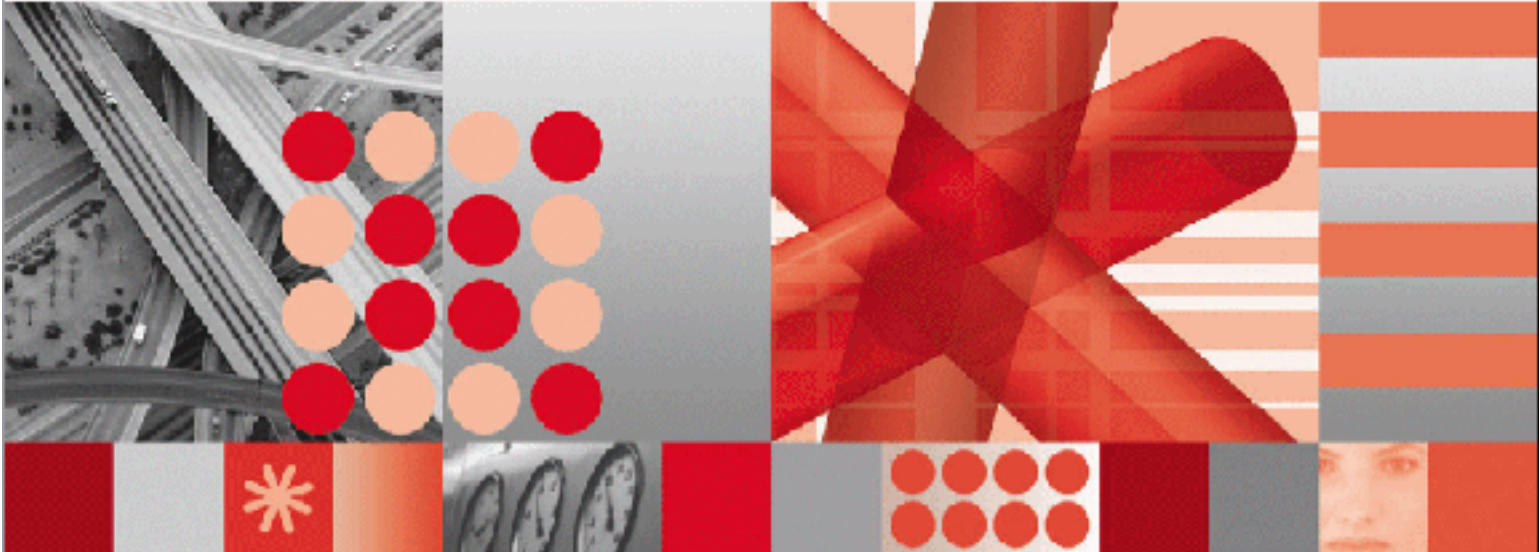




Releases 5 and 6



**Best Practices for Maximo System Performance**

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 55.

**Second edition, December 2008.**

This edition applies to Releases 5 and 6 of IBM Maximo and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2007, 2008. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>About This Publication</b> .....	<b>vii</b>
Intended Audience .....	vii
Related Information .....	vii
IBM Support .....	viii
<b>Chapter 1: Introduction</b> .....	<b>1</b>
Best Practices Is a Cooperative Effort .....	1
Quality Assurance and Testing .....	1
Factors in System Performance .....	2
The Maximo Infrastructure .....	2
Building Performance Improvements into Maximo .....	4
Performance Improvements in Release 6 .....	5
Bandwidth .....	5
Database .....	5
Integration and Batch Processing .....	5
<b>Chapter 2: System Architecture and Application Server Configuration</b> .....	<b>7</b>
Terminology .....	7
Basic Maximo System Configuration .....	8
Advanced Maximo Enterprise Configuration .....	9
Deploying Maximo in Multiple Clusters .....	9
Isolating Maximo User Applications .....	10
Isolating Reporting .....	10
Application Server Hardware Configuration .....	11
Application Server Tuning .....	11
HTTP Request Threads .....	11
The Correct Number of Threads .....	11
Managing Integration Load .....	12
Database Connection Pool .....	12
Default Connection Settings .....	12
Revising Connection Settings for a Clustered Environment .....	12
Load Balancing .....	13
<b>Chapter 3: Scheduled Tasks (Cron Tasks)</b> .....	<b>15</b>
Configuring Cron Tasks to Specific Maximo Application Servers .....	15
Configuring Cron Tasks in Release 6 .....	15
Configuring Specific Cron Tasks to Run on a Specific Maximo Application Server .....	16
Configuring Cron Tasks in Release 5 .....	16
<b>Chapter 4: Integration</b> .....	<b>17</b>
Integration and Batch Processing, Release 6 .....	17
Performance Issues in Integrated Systems .....	18
Clustering with the Maximo Enterprise Adapter .....	18
Recommended Approach to Clustering .....	18
Using Exception-Handling Queues .....	19
Using a Loop-Back Exception Queue Design .....	19
<b>Chapter 5: Reporting</b> .....	<b>21</b>
Run Resource-Intensive Reports in Off-Peak Hours .....	21

Limit the Use of Reports . . . . .	21
Run Reports on a Separate Cluster . . . . .	22
Provide a Separate Database for Reporting . . . . .	22
Manage Your Reports . . . . .	22
Configure the Actuate Report Server if Necessary . . . . .	23
<b>Chapter 6: Tuning the Database and Tuning SQL . . . . .</b>	<b>25</b>
The Maximo Database . . . . .	25
Database Tuning . . . . .	25
Indexing . . . . .	25
Special Index Types . . . . .	26
Statistics . . . . .	26
Customization . . . . .	26
Load Testing . . . . .	26
Oracle Database Notes . . . . .	26
SQL Server Database Notes . . . . .	26
User Queries . . . . .	27
Setting the Appropriate Search Type . . . . .	27
EXACT Search Type . . . . .	27
TEXT Search Type . . . . .	27
WILDCARD Search Type . . . . .	28
NONE Search Type . . . . .	28
Resetting the DATETIME Maximo Data Type of a Column . . . . .	28
Determining What Users Will Query . . . . .	29
Restricting Querying in Applications . . . . .	29
Key Performance Indicators . . . . .	29
Live KPIs on Start Centers . . . . .	30
KPI Queries . . . . .	30
KPI Cron Task Frequency . . . . .	30
KPI Best Practices . . . . .	30
Escalations . . . . .	30
Efficiency and Frequency . . . . .	30
Reports . . . . .	31
Turning Off Result Counts in Maximo Database Queries . . . . .	31
Release 5 . . . . .	31
Release 6 . . . . .	31
<b>Chapter 7: Network and Bandwidth . . . . .</b>	<b>33</b>
Maximo on a Network . . . . .	33
Using Citrix or Windows Terminal Server . . . . .	33
GZIP Capability . . . . .	34
Image and JavaScript Browser Caching . . . . .	34
Use Quality-of-Service Guarantees for Maximo Traffic . . . . .	34
Network Appliances . . . . .	35
<b>Chapter 8: Client Workstation Configuration . . . . .</b>	<b>37</b>
Workstation Configuration . . . . .	37
Configuration Recommendations . . . . .	37
<b>Chapter 9: Performance Improvement Tips and Customer Suggestions . . . . .</b>	<b>39</b>
Apply the Latest Patch . . . . .	39
Client Workstation Suggestions . . . . .	39
Remove Browser Controls from the User Interface . . . . .	39
Limit or Prevent Some Workstation Activities and Processes . . . . .	39
Monitor the Network for Streaming Audio and Video . . . . .	40
Have Only One Network Link Active . . . . .	40
Monitor for Hung Processes and Applications . . . . .	40
Garbage Collection Parameters for WebSphere . . . . .	40

Cross-site Scripting Filter Is Disabled by Default . . . . .	40
Limiting Use of E-Audit . . . . .	41
<b>Chapter 10: Troubleshooting . . . . .</b>	<b>43</b>
Documents or Files to Collect for Troubleshooting . . . . .	43
Troubleshooting Performance Problems . . . . .	44
Setting Up a Stand-alone Application Server for Debugging . . . . .	44
Debugging Parameters . . . . .	44
Displaying Garbage Collection Statistics on the Server . . . . .	45
Enabling Debug Mode for Maximo Enterprise Adapter Integration Logging . . . . .	46
Troubleshooting Performance Issues in Application Server Configuration. . . . .	46
Monitoring the Issues . . . . .	46
Addressing the Issues. . . . .	47
<b>Appendix A: Setting Up Two Clusters . . . . .</b>	<b>49</b>
Overview. . . . .	49
Example: Setting Up the UI Cluster and the Q Cluster . . . . .	50
Steps to Set Up the UI Cluster and the Q Cluster . . . . .	50
Result of Setup . . . . .	51
Accessing the Sequential Queue . . . . .	52
Example Deployment and Configuration. . . . .	52
<b>Notices . . . . .</b>	<b>55</b>



# About This Publication

This section explains how this guide can help you to use IBM<sup>®</sup> Maximo<sup>®</sup>. It also provides information about other IBM Corporation resources available to you, such as additional documentation and support.

## Intended Audience

This document is intended for people with system administration skills who are responsible for deploying Maximo in an enterprise setting and who want to optimize system performance.

## Related Information

For more information about IBM Maximo, refer to the following documentation:

Document	Description
<i>IBM Maximo Finance Manager's Guide</i>	Describes how IBM Maximo completes financial transactions and how to set up general ledger accounts.
<i>IBM Maximo Installation Guide</i>	Describes how to install and configure the following software: <ul style="list-style-type: none"><li>▼ Application server</li><li>▼ IBM Maximo</li><li>▼ Actuate<sup>®</sup></li></ul>
<i>IBM Maximo Multisite Administrator's Guide</i>	Describes how to configure IBM Maximo for a Multisite implementation.
IBM Maximo Online Help	Provides step-by-step procedures for IBM Maximo applications. Online help has more detailed procedural information than the <i>User's Guide</i> .
<i>IBM Maximo Reconciliation Module Implementation Guide</i>	Describes how to use the IBM Maximo Reconciliation module to reconcile the two types of information that IBM Maximo maintains about information technology (IT) assets: <ul style="list-style-type: none"><li>▼ IT asset data</li><li>▼ Deployed asset data.</li></ul>

Document	Description
<i>IBM Maximo Report Administration and Development Guide</i>	Describes how to use Actuate to design and administer IBM Maximo reports.
<i>IBM Maximo System Administrator's Guide</i>	Describes database configuration, security, and other administrative level applications and tasks.
<i>IBM Maximo User's Guide</i>	Provides an overview of the IBM Maximo end-user applications. It also describes how the IBM Maximo applications interact with each other.
<i>IBM Maximo Workflow Implementation Guide</i>	Provides information about how to use IBM Maximo to plan, design, build, test, implement, and manage Workflow processes.
<i>IBM Maximo Enterprise Adapter System Administrator's Guide</i>	Describes how to configure and use the IBM Maximo Enterprise Adapter.

## IBM Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. For information about the types of maintenance contracts available, see "Enhanced Support," in the *Software Support Handbook* at [techsupport.services.ibm.com/guides/services.html](https://techsupport.services.ibm.com/guides/services.html).

Complete the following steps to contact IBM Software Support with a problem:

- 1** Define the problem, gather background information, and determine the severity of the problem. For help, see "Contacting IBM" in the *Software Support Handbook* at [techsupport.services.ibm.com/guides/beforecontacting.html](https://techsupport.services.ibm.com/guides/beforecontacting.html).
- 2** Gather diagnostic information.
- 3** Submit your problem to IBM Software Support in one of the following ways:
  - a** Online: Click the **Report problems** tab on the IBM Software Support site: [www.ibm.com/software/support/probsub.html](https://www.ibm.com/software/support/probsub.html).
  - b** By telephone: For the telephone number to call in your country, go to the Contacts page of the *Software Support Handbook*: [techsupport.services.ibm.com/guides/contacts.html](https://techsupport.services.ibm.com/guides/contacts.html).

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM

Support web site daily, so that other users who experience the same problem can benefit from the same resolution.



# Introduction

# 1

IBM Maximo has a long and successful history in the world marketplace. Over the years, Maximo has incorporated many new features and grown in complexity. It also integrates with other complex software systems.

Small, medium, and large organizations implement Maximo in increasingly complex ways. For many customers, Maximo is now a global, enterprise-wide implementation used by thousands of users.

This book provides information to help you ensure good system performance. It addresses issues that affect performance in small, medium, and large deployments of Maximo.

The larger and more complex the deployment of IBM Maximo, the more challenging it is for you to keep Maximo performing well for end users. Because some of the greatest challenges are faced by those who deploy Maximo across large, global enterprises, this document has a special focus on improving performance in advanced enterprise configurations.

## **Release 5, Release 6**

This book addresses users of Maximo Release 5 and Release 6. You can implement many of the suggestions on either of these releases. Some ways to improve performance are for Release 6. When suggestions or best practices are available only on a specific release of Maximo, it is noted.

## ***Best Practices Is a Cooperative Effort***

This book is the result of a cooperative effort between IBM employees and a core group of Maximo enterprise customers. It was developed both in response to customers and in concert with them. The cooperative effort is ongoing.

System performance information that has been developed both within IBM and in actual deployments of Maximo is gathered here. The goal of this book is to provide you with information that can help you improve the experience of system performance for your users.

## **Quality Assurance and Testing**

The IBM quality assurance team does extensive testing of Maximo. We test during the development cycle, and after development is complete, before the product is released. Testing continues after Maximo is released.

## Factors in System Performance

Testing covers not only application and system functionality, but also system performance. We try to emulate many of the kinds of system setups that our customers might use.

We recognize, however, that a laboratory testing environment can never fully anticipate all of the scenarios that Maximo is put through.

For example, some customers have reported that testing with “only” one million work orders is insufficient to model the work order load of some global implementations. Efforts are now underway to increase the number and complexity of work orders used in our test scenarios.

The IBM Maximo testing environment undergoes constant review to determine how it best can be scaled to meet the demands of large enterprise deployments. Customer involvement and feedback is vital to improving our quality assurance testing.

## Factors in System Performance

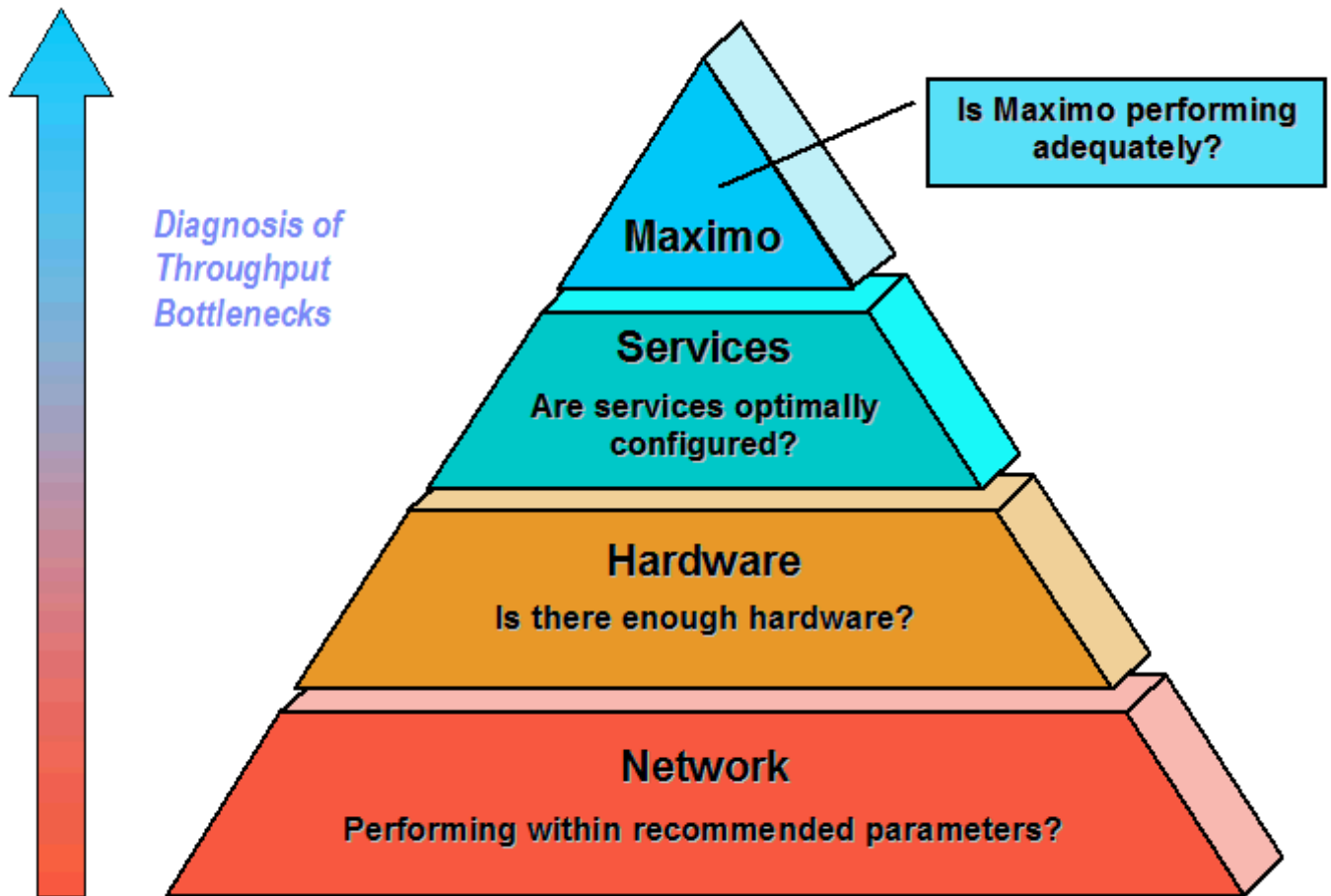
Maximo system performance depends on more than the Maximo applications and the database. The network architecture affects performance. Application server configuration can hurt or improve performance. The way that you deploy Maximo across servers affects the way that Maximo performs. Many other factors come into play in providing the end user experience of system performance.

Subsequent chapters in *Best Practices* address the following topics:

- ▼ System architecture setup
- ▼ Application server configuration
- ▼ Scheduled tasks (cron tasks)
- ▼ Reporting
- ▼ Integration with other systems with the Maximo Enterprise Adapter
- ▼ Network issues
- ▼ Bandwidth
- ▼ Load balancing
- ▼ Database tuning
- ▼ SQL tuning
- ▼ Client workstation configuration
- ▼ Miscellaneous performance improvement tips
- ▼ Troubleshooting

## The Maximo Infrastructure

The figures that follow illustrate the interdependence of Maximo and other elements that make up the entire system infrastructure.

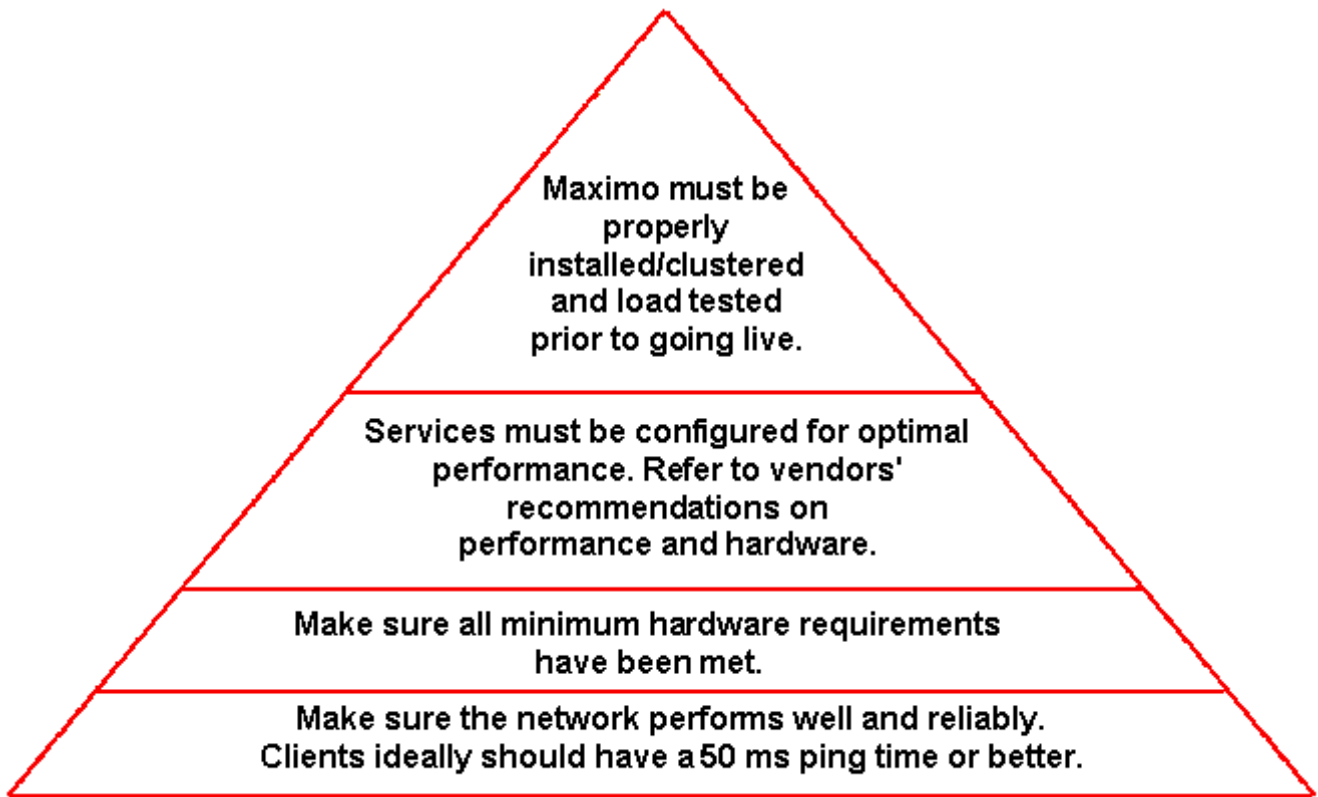
**Maximo Infrastructure Pyramid**

Maximo can be seen as sitting at the apex of a pyramid that consists of the network, hardware, and software services. This pyramid comprises the infrastructure within which Maximo operates.

- ▼ The entire Maximo infrastructure relies on the network architecture.
- ▼ Hardware servers operate within the network.
- ▼ Software services such as application servers, report servers, and database servers, operate on the hardware servers.
- ▼ Maximo runs within the sphere of the software services.

For Maximo to perform well, the entire system must be sufficiently robust and well tuned.

**Maximo Infrastructure Pyramid Tuning**



While achieving a 50-ms ping time is a challenge in many environments, you can use this target as a guideline.

Improving system performance generally requires attention to all elements of the Maximo infrastructure. Your databases can be tuned perfectly, but if the network is slow, you will have system performance issues. Or you might have a very fast local area network, but if client workstations lack memory and processing power, users will experience poor performance.

Customers report that there is normally not just a single issue they must address. Improving performance involves making adjustments to many things. Together, these changes can combine to yield significant performance improvement.

But you can “cover up” some performance issues by making improvements in only one or two areas. For example, upgrading end user workstations can often greatly improve the experience of system performance, even if other problems exist in the system.

## **Building Performance Improvements into Maximo**

With each new release of IBM Maximo, we provide new features for your end users. We also strive to increase performance reliability and speed with each new release. One way that we work to enhance performance is to provide new tools and features specifically aimed at improving system performance.

This section briefly notes some of the performance improvements that are incorporated into Maximo 6. Subsequent chapters provide more information about how to take advantage of the enhancements in your own setup.

## Performance Improvements in Release 6

Release 6 of Maximo provides performance improvement capabilities in a variety of areas.

### Bandwidth

Customers typically use commercial network appliances to optimize network performance. Network appliances such as Juniper, Riverbed, and others provide compression and caching features.

### GZIP Capability

GZIP is a compression utility that compresses the data traffic between the browser and the application server, which reduces bandwidth usage.

### Image and JavaScript Browser Caching

You can enable a browser file caching filter on the application server to store images, CSS files, and Java™Script files locally.

For more information about GZIP and browser caching, see Chapter 7, "Network and Bandwidth," on page 33.

### Database

#### Configurable Search Options

In Release 6, you can configure search options. You can change the default search type of WILDCARD to TEXT or EXACT. TEXT and EXACT searches can make use of indexes. Searches that use indexes are generally faster, especially on large tables.

For more information about configuring the search type, see Chapter 6, "Tuning the Database and Tuning SQL," on page 25.

### Integration and Batch Processing

#### Multi-threading

Maximo 6 employs a multi-threaded integration model. The Maximo Enterprise Adapter is set up to process multiple inbound transactions at once.

#### Cluster Support

Integration via the Maximo Enterprise Adapter supports the use of clustering.

Release 6 also provides cluster support for cron tasks.

Maximo Enterprise Adapter cron tasks can be set up so that they run on a specific cluster, or even on a specific Maximo application server.

For more information, see Chapter 2, "System Architecture and Application Server Configuration," on page 7. Also, see Chapter 3, "Scheduled Tasks (Cron Tasks)," on page 15. And, see Chapter 4, "Integration," on page 17.



# System Architecture and Application Server Configuration

# 2

This chapter addresses ways to help maximize system performance by configuring your system architecture to account for your resource needs.

## Terminology

This section provides a brief explanation of some terms that are used in this book.

An **application server** is software that maintains and provides an infrastructure to run applications (such as Maximo). IBM WebSphere® Application Server and BEA WebLogic are the two standard application servers that Maximo uses.

A **Maximo application server** maintains Maximo business objects and configuration files. Maximo application servers are created in WebSphere or WebLogic.

A **cluster** is a collection of application servers that perform the same task. For example, you can set up a user-interface cluster that is dedicated to serving end users who are using browsers. Each application server in the cluster has the same setup and configuration parameters.

A **JVM (Java Virtual Machine™)** is a platform-independent execution environment that converts Java code to machine language and executes it. Each application server runs on a different JVM. JVMs in a cluster can run on the same physical piece of hardware or on different hardware.

A **processor** is the central processing unit of the computer. A typical server has four processors, but it can have more. Set up one JVM for each processor.

**Scaling** is the process of adding hardware, adding memory, or adding processors to an existing system. With the ability to deploy business rules or components on multiple servers, Maximo provides virtually unlimited scalability.

**Vertical scaling** is the process of adding memory or processors to a computer to give it more processing power. **Horizontal scaling** is the process of adding hardware to the system.

**Front-end transactions** and **user load** are transaction requests that are initiated from a browser. **Back-end transactions** and **nonuser load** are processing requests that occur on a scheduled basis, or that come in by way of the integration components.

# Basic Maximo System Configuration

A basic Maximo configuration consists of IBM Maximo running on a single application server. The application server connects to a single instance of the Maximo database that is available on a database server. The application server also connects to a report server.

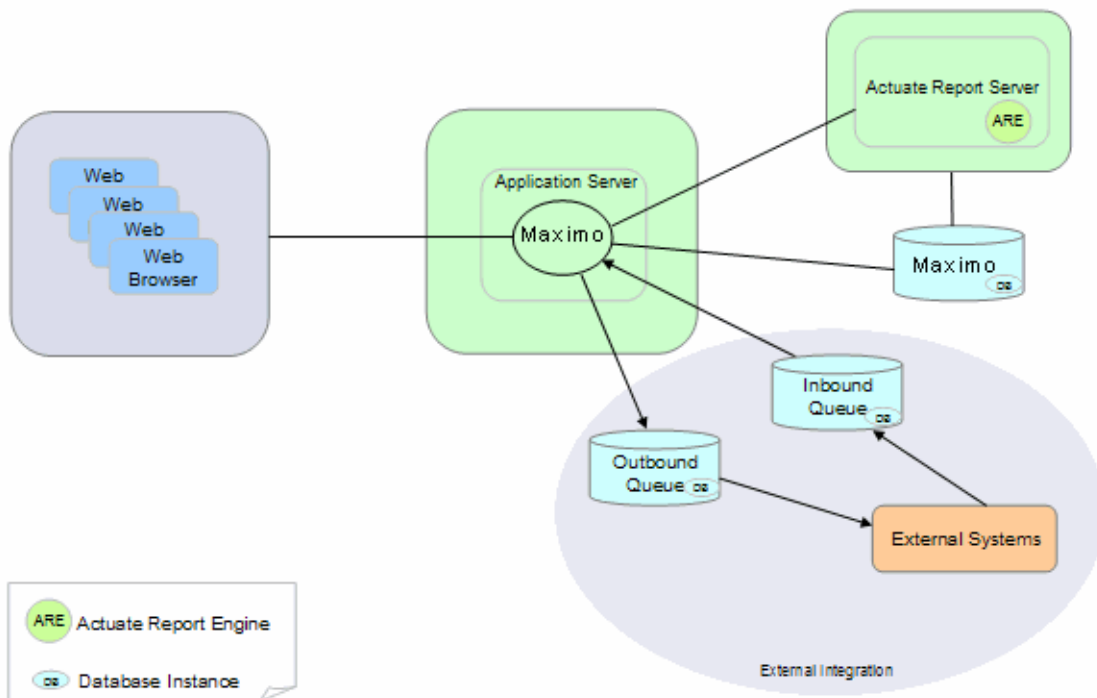
If the Maximo Enterprise Adapter is also configured for deployment, then additional messaging queues are set up. Maximo uses the additional queues to send data to external systems, and to receive data from external systems.

## Suitable for Up to 50 Users

Your testing and QA configuration environment (to test the development work) should closely simulate your eventual production environment. The basic configuration is appropriate for development configuration and for QA configuration if your production system is expected to have a user load of 50 or fewer users. And it is suitable for a production system with an actual user load of 50 or fewer users.

The following diagram illustrates the high-level components of this configuration.

### Maximo 6 Basic Configuration



Even with fewer than 50 users, the basic Maximo configuration can become overloaded if a great deal of processing is being done. For example, running scheduled jobs (cron tasks) and running reports both require considerable memory and processing power.

When processing power and memory are overtaxed, the end user who is using the application from a browser can experience unacceptable performance. If the

basic configuration is performing poorly, you might need to deploy the advanced enterprise configuration.

In the basic configuration, even if the application server process runs on a 64-bit computer, performance can be slow. Although a 64-bit computer allows greater memory utilization, the server process suffers from Java garbage collection pauses due to large heap space, and performance slows.

## Advanced Maximo Enterprise Configuration

If you have a small to medium-sized Maximo implementation, follow the standard Maximo installation and setup instructions and application server (WebSphere or WebLogic) configuration instructions. If your production system must support hundreds of users, the basic configuration will likely prove insufficient due to memory and processing limitations. A single instance of the application server to support the Maximo application cannot handle the load that is required in enterprise deployments.

Because Maximo is an interactive application, the end user who uses Maximo from a browser expects response from the server to be immediate, or nearly so.

Other processes in Maximo, such as cron tasks and inbound messages from external systems, do not require user interaction. Response time for these processes does not need to be immediate. These processes can be configured to run in separate clusters or on separate hardware.

## Deploying Maximo in Multiple Clusters

Use load distribution in a heavily used system that experiences unacceptable delays in Maximo applications and overall throughput. Load distribution separates the user-interface traffic from all other processing in Maximo.

Deploying Maximo on more than one server (or *clustering*) is an effective way to distribute user load and improve the end-user experience of system performance. When you use clustering, implement the cluster configuration so that the system can scale well.

You can set up multiple clusters, each consisting of multiple Java virtual machines (JVMs). The number of JVMs depends in part on the overall hardware and software limitations of the environment. You can tune the setup based on traffic and performance numbers.

A system of separate clusters with multiple JVMs as cluster members provides advantages in system administration. In such a setup, Maximo end users are not affected by any problem in the queue cluster or the Maximo cron task cluster that requires a cluster restart.

For details of a two-cluster setup, see Appendix A, "Setting Up Two Clusters," on page 49.

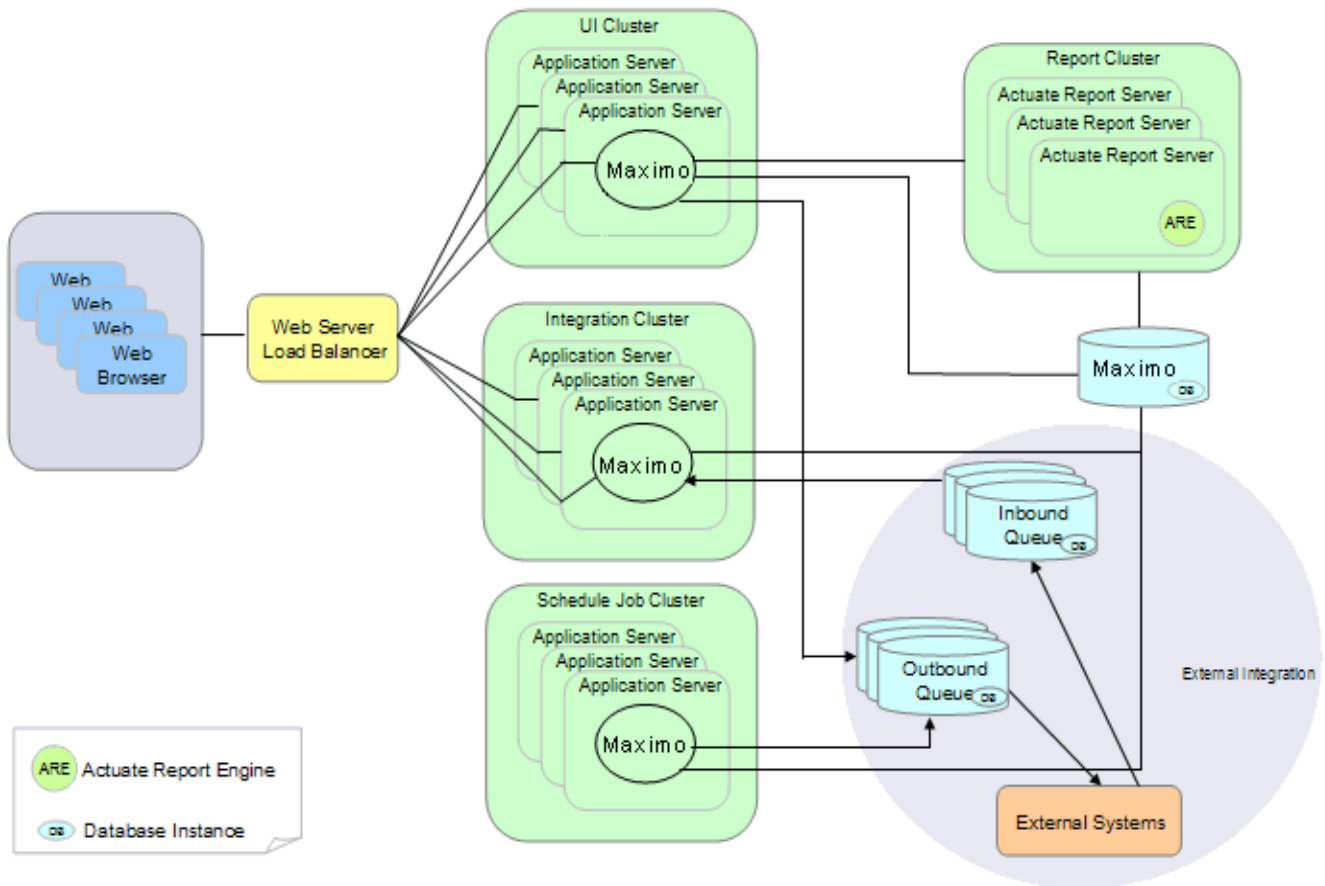
## Isolating Maximo User Applications

For your system to perform well, isolate the Maximo user-interactive applications to one cluster, and the asynchronous processes to a separate cluster. Such a configuration also helps the system to scale well.

You can further enhance each cluster simply by adding servers, depending on your needs.

The following diagram illustrates the high-level components of an advanced configuration.

### Maximo 6 Advanced Configuration



## Isolating Reporting

The use of Maximo application reports can also put a large burden on the Maximo user-interactive applications. To remove the impact of reports on the user-interface performance, run reports that require significant system resources as scheduled reports on separate clusters.

To help the overall system performance, run only simple reports from the user-interactive application.

For more information on managing the reporting function, see Chapter 5, "Reporting," on page 21.

# Application Server Hardware Configuration

Providing the following minimum hardware for each 50 concurrent users on the application server can help ensure good system performance.

- ▼ 1 800Mhz or faster processor
- ▼ 2 GB or more of memory

On a four-processor computer, dedicate one processor and 1 GB of memory to the operating system and application server. The typical four-processor server can support approximately 150 concurrent users.

For example, to support approximately 150 concurrent users at the minimum recommended configuration, you should have three Java virtual machines (JVMs). Each JVM should have a processor and 2 GB of memory dedicated to it.

## Application Server Tuning

Tune application server parameters that affect system performance.

### HTTP Request Threads

Application servers use threads to listen for and process requests from the browser. A thread represents a work item or task.

Application servers (WebSphere Application Server and BEA WebLogic) have default settings for number of threads.

- ▼ WebSphere Application Server has a default minimum thread size of 5.
- ▼ WebLogic has a default thread count of 25 (when started in production mode).

You can, and often should, revise these values to improve system performance. To optimize performance, you want to try to determine the right number of threads.

### The Correct Number of Threads

If the thread count is too high, you can have too many simultaneous requests to the CPU. Servicing the requests becomes inefficient.

If the thread count is too low, the processor is not fully utilized. The queue grows, and work backs up.

The correct number of threads varies from site to site. The number depends on the volume of front-end transactions (user load) and back-end transactions (nonuser load).

Use the WebSphere and WebLogic application server monitoring features to set and monitor thread usage and queue times for your application servers. An optimal number of threads can often be found in the 40 to 55 range.

## Managing Integration Load

If you are using the Maximo Enterprise Adapter, your system has front-end users plus potentially heavy resource use from integration. Integration volume can affect the front-end user experience of system performance.

One way to address potential performance degradation is to control resources used by integration. To do this, you can limit the number of message-driven beans (MDBs) on the continuous queue of the integration framework. (Another way to address performance is to separate inbound integration traffic to another server or cluster.)

A default installation of Maximo does not have a specific number of MDBs in the deployment descriptor file. The number of MDBs that gets created is managed by the container. Depending on the load on your system, you might need to reset this number. It is generally best to start off with a small number, such as two MDBs.

For a small or medium-sized implementation, two or three MDBs usually yields acceptable performance. As the system load increases, you might need to gradually increase the number of MDBs. You generally should not exceed ten MDBs. More than ten MDBs does not necessarily yield higher throughput and it can significantly increase resource usage (CPU cycles in particular).

Test different numbers of MDBs in a development setting to determine an appropriate number before you establish the number of MDBs for your production system.

## Database Connection Pool

Maximo uses proprietary connection pooling. If necessary, you can tune the connection pool settings for better management of connections.

### Default Connection Settings

By default, the MXServer.properties file (Release 5) and the maximo.properties file (Release 6) contain the following settings:

- ▼ mxe.db.initialConnections=15
- ▼ mxe.db.maxFreeConnections=30
- ▼ mxe.db.minFreeConnections=10
- ▼ mxe.db.newConnectionCount=5

The default connection settings are optimal when only one JVM is deployed. However, when Maximo is implemented in a clustered environment, these settings can create a large number of open connections to the database. A large number of unused connections results in unnecessary memory usage.

### Revising Connection Settings for a Clustered Environment

In a clustered environment, you can tune the connection settings for fewer open connections to improve performance.

The following revised connection settings might benefit database and system performance:

- ▼ mxe.db.initialConnections=6
- ▼ mxe.db.maxFreeConnections=10
- ▼ mxe.db.minFreeConnections=4
- ▼ mxe.db.newConnectionCount=3

Setting	Recommended value	Result
Initial connections	6	Each server opens six connections to the database when it starts up.
Minimum connections	4	If the connections fall below four, more connections are allocated. Initially, three user log-ins cause more connections to be allocated ( $6-3 = 3$ ).
New connections	3	When the connections drop below the minimum setting of four, three new connections are allocated. In the example scenario, these settings now result in six open connections ( $((6-3) + 3 = 6)$ ).
Maximum connections	10	As people log out, the connections are freed. When the number of free connections exceeds ten, connections are released.

## Load Balancing

Load balancing is the distribution of the task load across multiple instances of an application. User load comes from logged-in Maximo users. Nonuser load comes from such things as scheduled jobs (cron tasks) and Maximo Enterprise Adapter incoming transactions.

Distribute user load and nonuser load to different application servers or clusters.

For user load, software load balancers and hardware load balancers are available. A software load balancer is included with the Maximo system.

You can use a commercially available hardware device for hardware load balancing.



# Scheduled Tasks (Cron Tasks)

# 3

A cron task is a task scheduled to run at a specified frequency, such as every 10 minutes, or every 48 hours. The Reorder routine and PM generation are examples of Maximo cron tasks. Cron tasks run as background tasks and require no direct end user action.

When there are large numbers of cron tasks running at the same time that there is significant end-user load, response time for the end user can decline. To improve response time for the end user, you can set up cron tasks to run outside of the end-user environment.

Cron tasks can also take advantage of clustering. You can run cron tasks on multiple separate computers to improve throughput.

## Configuring Cron Tasks to Specific Maximo Application Servers

The procedure to set up cron tasks to run outside of the end-user environment is somewhat different for Release 6 and Release 5.

### Configuring Cron Tasks in Release 6

To set up cron tasks to run outside the end-user environment in Release 6, complete the following steps:

- 1** Copy and save the maximo.properties file under <Maximo\_root>/applications/maximo/properties.
- 2** In the maximo.properties file, look for either of the following keywords: "Cron Task Manager properties" or "mxe.crontask.donotrun".
- 3** In the maximo.properties file that is to be used in the end-user clustered environment, uncomment the property mxe.crontask.donotrun (remove the two forward slashes, "//" that precede the property).
- 4** Set mxe.crontask.donotrun as follows: mxe.crontask.donotrun=ALL. Using this setting ensures that the cron tasks will not run in the end-user clustered environment.
- 5** Save the maximo.properties file, and use it to build the Maximo EAR file. This Maximo EAR file is to be deployed in the end-user clustered environment.

## Configuring Cron Tasks to Specific Maximo Application Servers

- 6 Copy the altered `maximo.properties` file and save it as `maximo.properties.clustered`.
- 7 Edit `maximo.properties` (the original file): comment out `mxe.crontask.donotrun=ALL` (precede it with two forward slashes, `"/"/`). Using this setting ensures that the cron tasks will run in the environment where the Maximo EAR file with this `maximo.properties` file is deployed.
- 8 Build a Maximo EAR file with this `maximo.properties` file and deploy it to one or more servers outside the end-user clustered environment.
- 9 Save this `maximo.properties` file as `maximo.properties.crontask`.

If this non-end-user clustered environment consists of more than one server, each cron task runs on only one server. If any server shuts down, the cron task starts on another running server.

## Configuring Specific Cron Tasks to Run on a Specific Maximo Application Server

If you want to separate any cron tasks to run on a specific Maximo application server, edit the `maximo.properties` file.

In `maximo.properties` (the original file), edit the property `mxe.crontask.donotrun` to include the specific cron tasks. Use the format `mxe.crontask.donotrun=crontaskname.instanceName, crontaskname.instanceName, . . .`

The list of cron tasks following the property name is the list of cron tasks that are to be separated from this environment. Unlisted cron tasks will run in this environment.

## Configuring Cron Tasks in Release 5

In Release 5, you edit the `MXServer.properties` file to configure cron tasks. In `MXServer.properties`, there are entries for individual cron tasks; for example, `mxe.crontask.psd.workflow.EmailNotification`.

To set up cron tasks to run outside the end-user environment in Release 5, complete the following steps:

- 1 In `MXServer.properties`, comment out (precede with two forward slashes, `"/"/`) the cron task entries that you do not want to run in the end-user clustered environment.
- 2 Save the `MXServer.properties` file.
- 3 Build the Maximo EAR file and deploy it.
- 4 To enable the cron tasks in a non-end-user clustered environment, uncomment the individual cron tasks.
- 5 Save `MXserver.properties`.
- 6 Build the Maximo EAR file, and deploy it in the non-end-user clustered environment.

# Integration

# 4

This chapter provides information on improving system performance in a setting where the Maximo Enterprise Adapter is used to integrate with other systems.

## Integration and Batch Processing, Release 6

Multi-threaded integration was introduced in Maximo Release 6. Use of multiple threads allows the Maximo Enterprise Adapter to process multiple inbound transactions at one time.

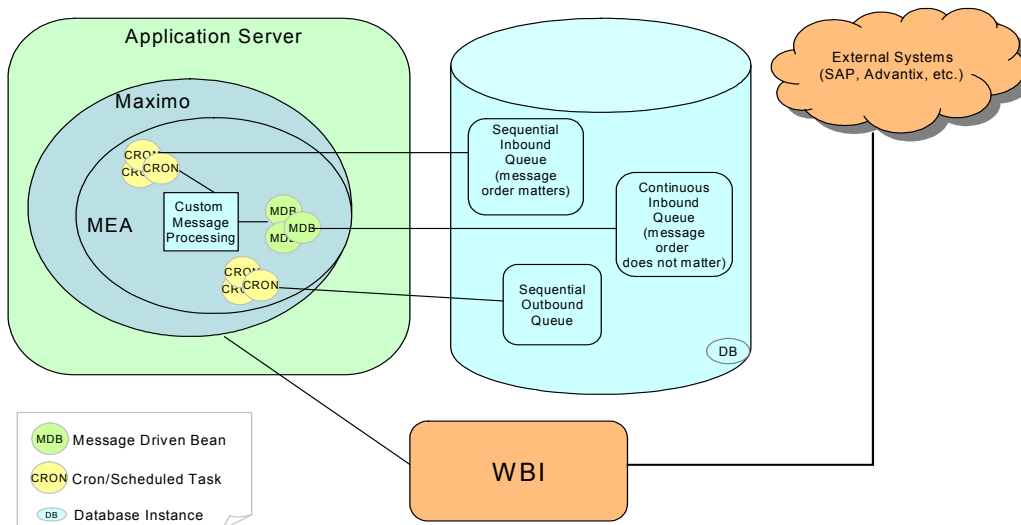
Maximo Release 6 also supports clustering. Transactions can be clustered across multiple physical computers.

Cron tasks can also take advantage of clustering. You can run cron tasks on multiple separate computers. Running cron tasks on separate computers improves throughput and also helps improve fault tolerance.

Incorporation of both multi-threading and clustering greatly increases throughput in an integrated environment.

### Overview of Integration

### Logical Overview - Integration



# Performance Issues in Integrated Systems

In an integrated Maximo environment, one of the main causes of falling performance numbers is the demand of non-user-load processes via the Maximo Enterprise Adapter.

The continuous queue functionality that the Maximo Enterprise Adapter uses is resource-intensive if there is a large amount of queued data. Queueing can require significant memory and lots of CPU cycles.

In a large, integrated system, you need to tune the Maximo parameters and the overall environment parameters, including the parameters for the application server.

## Clustering with the Maximo Enterprise Adapter

Multiple cluster setup is a good idea when Maximo integration with external systems (the Maximo Enterprise Adapter) is implemented. In a typical implementation, Maximo is connected with multiple external systems through Maximo Enterprise Adapter JMS queues.

In addition to setting up multiple clusters, you should determine and set appropriate cron task frequencies.

The purpose of clustering is to separate the inbound queue to a separate cluster from the interactive users. Separating the queue this way prevents inbound traffic from affecting the user interface performance.

Set up separate sequential inbound and sequential outbound queues for every system that Maximo communicates with. Using multiple queues prevents a problem in one system from affecting other unrelated traffic. If you also set up each sequential queue with its own cron task, administering the queues is easier.

## Recommended Approach to Clustering

One approach to clustering is to set up a system with three clusters. In this example, the clusters are name UI, Q, and Cron.

- ▼ UI cluster: provides connection for Maximo application end users. Hosts the Web application and sequential outbound queues.
- ▼ Q cluster: principally for hosting continuous and sequential inbound queues.
- ▼ Cron cluster: where other Maximo cron tasks are configured.

The UI cluster is where users generate transactions when they save work in Maximo. The UI cluster should therefore host sequential outbound queues to send out user-generated Maximo transactions to external systems.

The Q cluster is intended for incoming data for Maximo. However, there might be a situation when you also need sequential outbound queues in this cluster if outbound data gets generated as a result of inbound data. For example, a data loop-back condition might be initiated when an incoming receipt generates an inventory transaction that needs to be sent from Maximo. This loop-back behavior

is not standard in Maximo. You can, however, customize the system to work this way.

The optional Cron cluster is set up for other Maximo cron tasks. If those cron tasks might generate outbound Maximo transactions, then the Cron cluster needs its own set of sequential outbound queues.

For details on setting up a such a clustered environment, see Appendix A, "Setting Up Two Clusters," on page 49.

## Using Exception-Handling Queues

You can use exception-handling queues when system performance is being affected by bad data.

Bad data in a queue can come in two forms.

- ▼ The data needs manual intervention to correct.
- ▼ The "bad" data becomes "good" data in a matter of time or sequence.

An example of bad data that can become good data is a PO that specifies a vendor that is not yet saved in Maximo. After the vendor record is saved, the PO can be processed.

An effective way to deal with a high volume of this kind of bad data is to establish an exception-handling queue. You can sidetrack the bad messages from the continuous queue to another queue after a preset number of retries in the continuous queue. Sidetracking bad messages prevents the continuous queue from continuing to consume system resources for the bad data. The bad data can be processed in the second queue in a way that suits your own needs.

The Maximo Enterprise Adapter does not use a separate exception-handling queue. If you decide to implement such a queue, establish a business process and mechanism to reprocess corrected data from this queue.

## Using a Loop-Back Exception Queue Design

One approach to addressing a high volume of continuous queue errors is to take advantage of the exception queue concept of WebSphere. One design option is to use two exception queues:

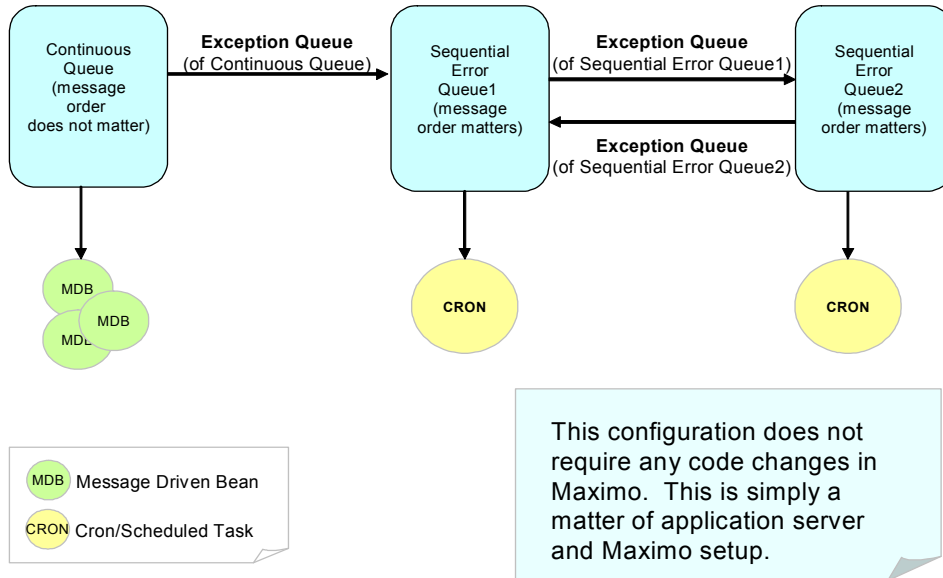
- ▼ Create an exception queue "A" for the continuous queue.
- ▼ For exception queue A, create its own exception queue, queue "B."
- ▼ Then specify that queue A is the exception queue for queue B.

This setup results in a loop-back or "figure eight" configuration between the two exception queues (which are set up as sequential queues). A bad message can get reinserted into a queue behind the message that allows it to become a good message. This setup allows most errors that are based on bad timing to be resolved without manual error correction.

If you set up an exception queue to handle errors, the error XML files are saved in the continuous queue folder, not in the exception queue folder hierarchy.

Loop-Back Queue Design

# Error Handling – Loopback (Figure 8) Queue Configuration



This chapter provides information on improving system performance by using best practices in reporting.

Running reports is a very resource-intensive operation and has the potential to be one of the biggest factors in poor system performance.

To help keep reports from affecting your system performance, you should isolate the reporting function as much as possible.

- ▼ Isolate reporting in time.
- ▼ Isolate reporting by user.
- ▼ Isolate reporting by cluster and application server.
- ▼ Isolate reporting by database.
- ▼ Manage the report database.
- ▼ Configure the report server.

## Run Resource-Intensive Reports in Off-Peak Hours

Many reports that consume significant resources are not needed immediately, and do not necessarily need to be run on up-to-the-minute data. You should run such reports in off-peak hours, such as overnight or on the weekend.

Time-based reports such as end-of-month and end-of-quarter reports can be run in off-peak hours, on a copied database from a specific date and time. Because you do not need to run these reports on the current database, you can protect the production database from being slowed by these reports.

## Limit the Use of Reports

The more users that run reports, and the more reports they run (especially database-intensive reports), the greater the potential effect on system performance. You should establish business practices to help manage the amount of report use, especially during peak system-use hours.

Limit the number of users who can run reports. Limit the number of reports that users can run.

During peak business hours, try to limit report use to reports that users need for their daily work, such as Print Work Orders, Print POs, and so on.

# Run Reports on a Separate Cluster

If your users do extensive reporting, a good practice is to establish one or more application servers that are dedicated to running reports. You can size the clustering of report application servers based on demand.

Establish a separate cluster for running scheduled reports (cron jobs).

## Provide a Separate Database for Reporting

Some customers report that *providing a separate database to run reports on is the single practice that gives the greatest boost to system performance.*

Configure a separate Maximo database that has a copy of the production data, and use that as an off-line database for reporting. Mirror the Maximo production database on a separate database server, and run resource-intensive reports on the mirror database.

Create a separate Maximo application that connects to the reporting database and synchronize the production and reporting databases periodically. For example, you might synchronize the databases at the end of every day or once a week, depending on your needs.

With this setup, reports that require more system resources can be run by just a few administration users. Because they are run on a separate mirror database, these reports do not affect performance of the production system.

## Manage Your Reports

By default, executed reports are saved to the Actuate Encyclopedia. Over time, the volume of saved reports can affect report performance. It is a good practice to periodically delete unneeded executed reports from the Encyclopedia.

You can delete unneeded reports from the Encyclopedia by enabling the Actuate AutoArchive feature in the Management Console.

AutoArchive sweeps the Encyclopedia for documents that are older than a specified age and deletes them.

For more information about the AutoArchive feature, see the *IBM Maximo Report Administration and Development Guide*.

# Configure the Actuate Report Server if Necessary

By default, the Actuate report server is configured for typical usage. The basic single-server setup is typically enough to support 100 users.

Actuate is a resource-intensive application. Allocate a minimum of two processors and 2 GB of memory to run Actuate.

If you are running a large number of reports, consider a load-sharing configuration. (See Chapter 2, "System Architecture and Application Server Configuration.") You can configure Actuate in a client/server configuration to support more users.

## Configure the Actuate Report Server if Necessary

# Tuning the Database and Tuning SQL

# 6

This chapter provides information on improving system performance by focusing on two related areas:

- ▼ Database tuning
- ▼ User queries and SQL tuning

## The Maximo Database

The database is central to Maximo functionality. The database stores all data that is collected and calculated by the Maximo application. The database also stores metadata for configuring and maintaining the environment.

The database server processes all transactions from the Maximo application. The integrated reporting function accesses the data in the database to generate documents such as work orders and purchase orders. Reporting also generates resource-intensive management reports.

Because all functionality is based on database performance, the database should be a key focus for performance tuning.

## Database Tuning

You can apply standard database-tuning techniques to Maximo. Periodically monitor a production Maximo database during peak load. You can use any appropriate monitoring tools or techniques. If necessary, adjust parameters to resolve bottlenecks that are reported by the monitoring tools.

## Indexing

Indexing a database requires a good understanding of the data, user functions, and how databases use indexes. Indexes use key parts of data from a table in a binary structure to enhance searching capability. Each record of data in the table must have associated data in the index.

Indexing can greatly increase search speeds. However, a drawback of indexes is that for each insert, update, or delete, the index must also be updated. Database administrators often apply many indexes to a table to enhance searching, and then find that other activities have slowed. You should review indexes to ensure that you have the right balance for searching and updating tables.

## Special Index Types

Some special index types are available on each database platform that are not available in the Maximo Database Configuration application. These index types can be created and maintained from the back end, and they can improve performance in specific cases. For example, on Oracle you might create a bitmap index or a function-based index if you determine that these indexes would improve certain queries.

If you use special index types, the Maximo system administrator must remember to remove any special indexes before configuring database changes. After the database is configured, the special indexes must be replaced.

## Statistics

Keep index statistics up-to-date. In the Maximo Database Configure application you can use a Select Action menu action that does this. You can also update index statistics from the back end.

With Maximo Release 6, the Update Statistics action is enhanced to function on Oracle by running the `MAXIMO_GATHER_TABLE_STATS` procedure.

## Customization

Customizing Maximo can change the way information is selected from the database. Some customizations include additional tables and columns. If you have customized Maximo, you should carefully compare indexes to the user functions that use them. Ensure that you implemented the right balance of indexes.

## Load Testing

If possible, do load testing during the implementation phase to expose performance problems before you put Maximo into production. If you have the equipment to perform load testing, you can use load testing after Maximo is in production to determine if there is any performance impact from patches or from data growth over time.

## Oracle Database Notes

On Oracle, you should specify the following two initialization parameters:

- ▼ Set the **Cursor\_sharing** parameter to `SIMILAR` or `FORCE` so that the user-entered literal values are converted to bind variables.
- ▼ Set the **Nls\_length\_semantics** parameter to `CHAR` when the database character set is a double-byte or unicode character set.

## SQL Server Database Notes

Microsoft® SQL Server poses some challenges in environments like Maximo that have multiple concurrent-user transactions. When records are selected, SQL Server escalates locks from "record" to "page" to "table," based on an internal

algorithm. If user queries are not efficient, database locks can impede other users who are accessing the system.

Some customers running Maximo on SQL Server have improved database performance by turning on the SQL Server row-level locking that is supported by Microsoft. Turning on row-level locking reduces the locked areas of the database. But row-level locking also can result in a higher number of locks, because locks are based on record instead of memory page. A document that describes how to implement row-level locking for Maximo on SQL Server is available at the following link:

[http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=%22row+level+locking%22&uid=swg21261979&loc=en\\_US&cs=utf-8&lang=en](http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=%22row+level+locking%22&uid=swg21261979&loc=en_US&cs=utf-8&lang=en)

## User Queries

Most of the WHERE clauses in Maximo queries are generated by individual users on the List tabs of Maximo applications. This powerful feature of Maximo can produce a lot of inefficient SQL. You can improve the ease of use and convenience for users by setting the appropriate search types for database columns. Using appropriate search types also reduces the load on the database.

## Setting the Appropriate Search Type

In Releases 6, you can configure search options. You can change the default search type of WILDCARD to TEXT or EXACT. TEXT and EXACT searches can use indexes. You specify the search type for a database column in the Database Configuration application. You also can change the search type for groups of columns. See <http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&uid=swg21261876>.

### EXACT Search Type

When an end user does a search, the default method of searching on many Maximo database fields is to use wildcards (SEARCHTYPE = WILDCARD). The WILDCARD search type causes Maximo to construct a condition of the form

```
column like '%value%'
```

when the user enters "value" in a field on the List tab. In wildcard searching, the database engine cannot use indexes. Searching without indexes can result in slower search times, especially on tables with a large number of rows.

You can specify a search type of EXACT when word searching is not needed. For example, key fields (such as Work Order and Purchase Order) and value list fields (such as Work Order Status) can benefit from the indexing that is used in EXACT searches. EXACT searches use wildcarding only if a user explicitly enters wildcard characters on the List tab or in the WHERE clause.

### TEXT Search Type

Most Maximo tables have one or more longer character columns for descriptions, memos, or remarks. You can provide a search type of TEXT, and a corresponding

Oracle Text index or SQL Server full text catalog, for columns that have a lot of text. (Full text indexing is not available with Maximo on IBM DB2®.) Text indexing puts some load on the database because of the constant need for background processing to keep the indexes synchronized. However, text indexing produces efficient word searching of description fields.

Specify a search type of TEXT on description fields that word searching is required for. Use TEXT on description fields of tables with large numbers of rows (tens of thousands, for example).

The text search engine takes some time to refresh the indexes, so new records might not be found until the text search index refreshes itself.

On Oracle, you can modify the procedure `maximo_ts_job_call` to change the schedule of the synchronization process to any interval that you want.

On SQL Server, you can set and modify the population schedule for the Full Text Catalog. Use SQL Server 2000 Enterprise Manager or SQL Server 2005 Management Studio.

## WILDCARD Search Type

Tables with fewer than 2000 or 3000 records are often completely scanned regardless of indexes. The I/O cost to read the entire table is less than the average I/O cost of the index lookup plus the table lookup. The SEARCHTYPE value has no effect on database behavior when such scans are performed.

You can use the default search type of WILDCARD on description fields of tables that have a relatively small number of rows (2,000 or fewer, for example). Wildcard searching provides more flexibility for the end users. Tables with relatively few rows will have no noticeable degradation in performance.

## NONE Search Type

The NONE search type prevents a column from being used in a WHERE clause, and still allow display of the returned values on the List tab.

## Resetting the DATETIME Maximo Data Type of a Column

When you search on fields whose Maximo data type is DATETIME, the resulting database query uses TO\_TIMESTAMP in the WHERE clause. Queries that use TO\_TIMESTAMP are time-consuming.

You can edit the WHERE clause manually and replace TO\_TIMESTAMP with TO\_DATE. Queries that use TO\_DATE in the WHERE clause yield a faster search. However, you must edit the query each time you search on a DATETIME field.

In many cases, a better approach is to change the data type of routinely queried DATETIME fields to the data type DATE. Fields with a data type of DATE use TO\_DATE in the WHERE clause.

Use the Database Configuration application to change the data type of a column.

## Determining What Users Will Query

Maximo users usually have a well-defined set of columns that they want to query in each application. You can identify these columns in user team interviews during implementation, or later, by examining reports of slow-running SELECT statements. You can then index these columns to improve system performance.

Users can create and save their own queries, and can also share queries with other users. Saved queries are stored in a table named QUERY. You should periodically review these saved queries for inefficient conditions and use of unindexed columns.

Someone with SQL expertise can help create special-purpose queries (for example, return all PM work orders created since Monday of this week). Doing so can save end users the effort of querying larger sets and sorting and scrolling through them.

You also can set up users with an efficient default query for their most-used applications so that they see their preferred record set when they enter the application. For example, in Work Order Tracking, you might specify a default query for supervisor Smith so that he would initially see only work orders with SMITH in the Supervisor field.

## Restricting Querying in Applications

You can control or restrict user access to query features and user ability to query on specific columns by using a combination of methods:

- ▼ Application Designer
- ▼ Application cloning
- ▼ Security groups

### Application Designer

You can use the Application Designer to customize an application by adding or removing columns from the List tab. You can then ensure that the columns to be queried are all indexed.

### Application Cloning

You can clone an application and then use the Application Designer to create an alternate version of an application that has a restricted number of columns that can be queried.

### Security Groups

After you clone applications, you can use security groups to assign users to specific application clones.

You can also use security groups to prohibit access to the More Search Fields and Where Clause advanced query options. By prohibiting access to those options, you limit users to querying on the List tab of the application.

## Key Performance Indicators

Key performance indicators (KPIs) display the state of systems and processes in Maximo. Because KPIs can be user-defined, you should monitor them for efficiency.

## Live KPIs on Start Centers

When a KPI is defined on a Start Center, you have a choice of how to retrieve information. You can run an immediate query to get the information, or you can retrieve the information from a table that is updated by a cron task.

The immediate query runs every time the Start Center opens. This approach can cause a long delay in opening the Start Center, and it puts a load on the database.

## KPI Queries

You should periodically review queries that are written for KPIs. Check for SQL efficiency and index usage.

## KPI Cron Task Frequency

How up-to-date does your KPI information need to be? Five-minute intervals might sound like a good idea initially, but 60-minute or 120-minute intervals might be as useful to the people who want to see the information. Longer KPI cron task intervals reduce the load on the database from KPIs and can improve system performance.

In general, use longer rather than shorter KPI cron task frequency intervals when the value of the data to the end user is essentially the same at longer intervals.

## KPI Best Practices

- ▼ Set all KPIs to retrieve their data from the KPI table.
- ▼ Use cron tasks to run KPI queries at reasonable intervals.

For information on moving cron tasks to a separate server, see Chapter 3, "Scheduled Tasks (Cron Tasks)," on page 15.

## Escalations

Escalations are, in effect, batch users that do not have the overhead of a user interface. But escalations are very resource intensive whenever they do something.

## Efficiency and Frequency

An escalation selects a set of records and performs a list of one or more actions on the result set. The columns in the WHERE clause of the selection should be efficiently indexed.

Set the frequency or schedule of an escalation according to its importance:

- ▼ An escalation that dispatches emergency work orders for critical safety issues might warrant a frequency of five minutes.
- ▼ An escalation that ensures that service requests from executives are dealt with promptly might need to run every fifteen minutes.
- ▼ An escalation that closes work orders that were completed 90 or more days ago might need to run only once a week at an off-peak time.

## Reports

Review custom reports for efficient SQL and use of indexes. Most reports receive a WHERE clause from Maximo. For all these cases, improving the efficiency of user queries also improves the efficiency of reports.

## Turning Off Result Counts in Maximo Database Queries

By default, Maximo counts and displays the number of records that match a query. Counting large result sets can slow system performance. You can disable result-set counting.

### Release 5

In Release 5 of Maximo, disabling result-set counting is a global action. Editing the `jspsettings.txt` file affects all tables.

To disable result-set counting, complete the following steps:

- 1 Edit the `jspsettings.txt` file in `<Maximo_home>\resources\defaults\jspsettings.txt`.
- 2 Set `tabledisplaycount=false`.
- 3 Rebuild the EAR file and redeploy it.

### Release 6

In Release 6, you disable result-set counting on a table-by-table basis. You can disable result-set counting only for the tables with queries that are slow. You can leave result-set counting in place for tables whose queries do not slow system performance.

### Prerequisite

You must be on Release 6.0, Patch 02 or a later version of Release 6 to disable result-set counting.

To disable result-set counting, complete the following steps:

- 1 Edit the presentation XML files in `<Maximo home>\resources\presentations`.
- 2 In the table tag, add `showcount="false"` after `id="results_showlist"`.

### Example

In `wotrack.xml`, add `showcount="false"` to the table tag.

Table tag after editing:

```
<table id="results_showlist" showcount="false" datasrc="results_showlist"
mboname="WORKORDER" inputmode="readonly" selectmode="multiple"
label="Work Orders">
```

Complete the following steps to apply your changes:

- 1 Log in to Maximo as `maxadmin`.

## User Queries

- 2 Open the Application Designer.
- 3 Click **Import Application Definition**.
- 4 Browse to the edited presentation XML file.
- 5 Click **OK**.
- 6 Repeat steps 4 and 5 for each edited presentation XML file.
- 7 Restart the Maximo server.

# Network and Bandwidth

# 7

This chapter discusses approaches to improving system performance by focusing on network issues and bandwidth.

## Maximo on a Network

Clients connect to the Maximo application over the network. The application also communicates with its various parts (application server, database, report server) over the network. If any segment of the network performs poorly, the end user experiences a system that is slow and hard to navigate.

Maximo is a Web-based product that operates on a request and response basis. If the requests and responses are delivered slowly, Maximo itself has no control over response time.

Optimum network configurations for Maximo should include the ability to produce 50 ms or faster round-trip packet response between the client and the server. The system needs enough bandwidth to support 6 kbps per user. Users might begin to experience performance degradation if the network does not operate within these parameters.

## Using Citrix or Windows Terminal Server

You can use network caching, acceleration, and compression utilities to improve network performance.

Other options for resolving low bandwidth or high-latency network performance issues include services such as Windows<sup>®</sup> Terminal Server and Citrix. These services can help provide maximum performance between the Windows Terminal Server or Citrix client and the Maximo server with a minimum of traffic between the Windows Terminal Server or Citrix server and the end user.

A benefit of running Maximo through Citrix is that Citrix traffic is treated as “business traffic.” In some customer environments, business traffic can take priority over nonbusiness traffic.

Some Maximo customers have found considerable improvements in network performance when they use Citrix or Windows Terminal Server. Note that IBM does not test or certify Citrix or other bandwidth tools.

# GZIP Capability

GZIP is a compression utility that compresses the data traffic between the browser and the application server. Compressed data reduces bandwidth usage. GZIP is available in Release 6.

You must enable GZIP. For information about how to set up to use GZIP, see the following IBM Maximo Support Web page:

[Reducing Network Bandwidth Utilization via Response Compression and Browser Caching \(http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=cache&uid=swg21262009&loc=en\\_US&cs=utf-8&lang=en\)](http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=cache&uid=swg21262009&loc=en_US&cs=utf-8&lang=en).

The GZIP compression utility increases CPU utilization. You should enable GZIP only if low bandwidth is a factor in poor system performance.

## Image and JavaScript Browser Caching

You can enable a browser file caching filter on the application server. This lets the browser store images, CSS files, and JavaScript files locally. Storing images and files locally benefits performance in two ways:

- ▼ Images and files do not need to be constantly requested from the server.
- ▼ Because images are not downloaded as often, less bandwidth is required.

Browser file caching is not enabled by default. *We recommend that you enable browser file caching.* For information on how to enable this feature, see the following IBM Maximo Support Web page:

[Reducing Network Bandwidth Utilization via Response Compression and Browser Caching \(http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=cache&uid=swg21262009&loc=en\\_US&cs=utf-8&lang=en\)](http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&q1=cache&uid=swg21262009&loc=en_US&cs=utf-8&lang=en).

For Microsoft Internet Explorer 6, set the number of days until refresh is enabled to 32 days or longer. Modify the web.xml file. Set the max-age=<seconds> value to 2764800 or higher; 2764800 seconds equals 32 days.

## Use Quality-of-Service Guarantees for Maximo Traffic

You can prioritize Maximo traffic over standard Web traffic. Contact your network administrator for details about prioritizing Maximo traffic.

# Network Appliances

Many customers use network appliances, such as those provided by Juniper Networks, RiverBed, and Expand Networks, among other vendors.

Network appliances can help compress data and optimize bandwidth. Customers report that network appliances can prove very beneficial to system performance, especially in a high-latency environment.



# Client Workstation Configuration

# 8

For optimal performance of Maximo, you might need more than the minimum recommended hardware and software requirements.

This chapter provides recommendations for enhancing client workstation configuration. Consider implementing these recommendations if you suspect that the end user experience of system performance is at least partly attributable to the client workstation setup.

## Workstation Configuration

Some Maximo customers report that client workstation configuration is initially the most important area to focus on when users report performance issues.

If you provide users with adequate workstations that have sufficient memory, you will notice performance improvement.

## Configuration Recommendations

Consider using the following configuration for client workstations.

### Processor

Intel® Pentium® or Xeon® processor.

### Memory

At least 1 to 2 GB of RAM.

Although 512 MB of RAM has been a published recommendation, many customer sites find that this is insufficient for robust performance in an enterprise setting. Some customers report that users who had performance issues with 512 MB or 1 GB of RAM have fewer or no issues after increasing RAM.

### Monitor Resolution

Notebook screen resolution: 1024 x 768.

Desktop screen resolution: 1280 x 1024.

### Operating System

Microsoft Windows XP.

### Browser

Microsoft Internet Explorer 6 or 7. Use Internet Explorer 7 if the Maximo release supports Internet Explorer 7.

Some customers report noticeable performance improvements with Internet Explorer 7.

### File Reader

Adobe® Reader 6 or higher for PDF files.

## **Workstation Configuration**

### **Antivirus Software**

Install antivirus software on all client workstations.

Scanning for viruses can affect system performance. If possible, schedule virus scans during low usage periods.

# Performance Improvement Tips and Customer Suggestions

# 9

This chapter contains a variety of tips and suggestions that IBM Maximo experience and customers have shown can help improve system performance.

## Apply the Latest Patch

Apply the latest available patch, fix pack, or hot fix for your release of Maximo. Maximo patches and hot fixes contain fixes for application issues and often contain features that can improve system performance.

Patches can contain fixes specifically designed to improve performance. Patches also contain new features and parameters that help you to better monitor and debug system performance issues.

## Client Workstation Suggestions

This section contains suggestions for the client workstation that can improve system performance.

### Remove Browser Controls from the User Interface

Some customers have found that it is difficult to train end users to stop using browser control buttons. The Forward, Back, and Refresh browser controls can interfere with the way that Maximo Release 5 internal navigation works.

You can remove browser control buttons from the user environment. You can find the procedure by following this link:

[http://www-1.ibm.com/support/docview.wss?rs=3214&uid=swg21268053&loc=en\\_US&cs=utf-8&lang=en](http://www-1.ibm.com/support/docview.wss?rs=3214&uid=swg21268053&loc=en_US&cs=utf-8&lang=en)

### Limit or Prevent Some Workstation Activities and Processes

Customers report that some user activities and workstation processes can degrade performance. You should check for and monitor these activities and processes, and respond as necessary.

## Monitor the Network for Streaming Audio and Video

Customers report that monitoring the network in order to prevent users from using streaming audio and video can noticeably increase the bandwidth available to the Maximo system.

## Have Only One Network Link Active

If a user has both a wireless network link and a LAN link active, it can cause Maximo system performance issues. Limit users to one active network link.

## Monitor for Hung Processes and Applications

Processes and applications that are not responding (are hanging) use memory and can affect the performance of the client workstation. A workstation can have hung processes or applications that the user might be unaware of.

Use system tools, such as the Windows Task Manager, to check for and end hung processes and applications.

# Garbage Collection Parameters for WebSphere

On WebSphere, garbage collection for objects with remote references is not done in an optimal manner. To correct this, on the WebSphere application server under JVM properties, set the following parameters:

- ▼ Dsun.rmi.dgc.ackTimeout=100000
- ▼ Djava.net.preferIPv4Stack=true

For additional information on garbage-collection parameters for WebSphere, follow this link:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/urun\\_rconfproc\\_jvm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/urun_rconfproc_jvm.html)

# Cross-site Scripting Filter Is Disabled by Default

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in Web applications. Cross-site scripting can be used by malicious Web users to inject code into Web pages viewed by other users. Attackers can exploit the vulnerability to bypass access controls such as the same-origin policy.

A cross-site scripting filter is part of Maximo 5 and 6. The filter evaluates every user input for security vulnerability.

By default, the cross-site scripting filter is disabled. Enabling the filter causes some degradation in system performance.

Having the filter disabled is only a potential issue if you have applications deployed outside the firewall, on the public Internet. If you have Maximo applications deployed on the public Internet, you should enable the cross-site scripting filter.

# Limiting Use of E-Audit

E-audit is a feature that is designed to support industries, such as life sciences, that are required to track changes by user, date, and time. Because e-audit creates a transaction record for every change in the system, it has an effect on system performance. You should weigh the benefits of enabling e-audit against the impact on performance.

By default, when you enable e-audit for an object, such as the Workorder object, all fields are e-audited.

You should limit the number of objects that you e-audit. For each object that is e-audited, limit the number of fields that you e-audit.

Limiting the use of e-audit to only those objects and fields that require an audit trail limits the impact on system performance.



This chapter provides information on troubleshooting system performance issues.

Maximo 5 and 6 have features that log Maximo health statistics. In the most recent versions of Maximo, these features are turned on by default. These features and their associated data help you troubleshoot Maximo performance problems.

## Documents or Files to Collect for Troubleshooting

Before you troubleshoot or call IBM Support for system performance issues, collect the following information and documents or files.

- ▼ Version and release information for each Maximo product.
- ▼ Product, version, and release information for your database and application-server software.
- ▼ All application server log files. If you have multiple servers, collect logs for each server.
- ▼ Maximo log files. If you have multiple Maximo servers, collect logs for each server.
- ▼ Garbage collection information from the application servers. The application servers must be enabled for verbose garbage collection. For more information, see "Displaying Garbage Collection Statistics on the Server," on page 45.
- ▼ Maximo configuration settings, which are defined in the properties file, under applications/properties.
- ▼ If the problem is related to deadlock, generate five to ten thread dumps on the server instances at two-minute intervals.

- For information on generating thread dumps for WebSphere, follow this link:

<http://www-1.ibm.com/support/docview.wss?rs=0&uid=swg21138203>

- For information on generating thread dumps for WebLogic, follow these links:

<http://e-docs.bea.com/wls/docs81/perform/JVMTuning.html#1131866>

<http://edocs.bea.com/wls/docs81/cluster/trouble.html>

## Troubleshooting Performance Problems

- ▼ If the problem is an out-of-memory issue, collect the heap dump files associated with the server instance.
  - For information on collecting heap dump files for WebSphere, follow this link:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tprf\\_generatingheapdumps.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tprf_generatingheapdumps.html)

- For information on collecting heap dump files for WebLogic, follow these links:

<http://e-docs.bea.com/wls/docs81/perform/JVMTuning.html#1131866>

<http://edocs.bea.com/wls/docs81/cluster/trouble.html>

# Troubleshooting Performance Problems

Troubleshooting procedures are intended to be used in a development or test environment for performance analysis and debugging. The procedures generally should not be used in a production environment. Use these procedures in a production environment only if you cannot isolate the problem in a test environment.

## Setting Up a Stand-alone Application Server for Debugging

Debugging SQL or Maximo business object problems and using detailed logging when you reproduce other problems can generate large logs and slow the system.

You can set up a stand-alone application server for debugging. However, if issues that you are investigating cannot be replicated in a test environment, setting up stand-alone loggers provides a limited benefit.

Set the parameters so that the server does not execute cron tasks. The stand-alone application server can be on a separate computer. Using a separate computer lets you easily stop and start the server to change logging parameters and reproduce problems with a single user.

## Debugging Parameters

You can use debugging parameters with Maximo. In Maximo 6, the debugging parameters are configured in the Maximo.properties file. In Maximo 5, they are configured in the MXServer.properties file. The parameters have evolved as Maximo has evolved. They use different implementation criteria for different releases and patch levels. There are five debugging parameters:

- ▼ mxe.db.fetchResultLogLimit (5.2 P02A and later)
- ▼ mxe.db.logSQLTimeLimit (5.2 P02A and later)
- ▼ mbocount (5.2 P05 and later - different implementations)
- ▼ mxe.db.logSQLPlan (6.0 P01 and later; Oracle only)
- ▼ mxe.db.sqlTableScanExclude (6.0 P01 and later; Oracle only)

For information about implementing the debugging parameters, see the following IBM Support Web page:

[http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&uid=swg21291250&loc=en\\_US&cs=utf-8&lang=en](http://www-1.ibm.com/support/docview.wss?rs=3214&context=SSLKT6&uid=swg21291250&loc=en_US&cs=utf-8&lang=en)

## Displaying Garbage Collection Statistics on the Server

You can turn on garbage collection (GC) verbosity to find out how the garbage collection routine is functioning on the Maximo application server. Set the following parameter in WebSphere or WebLogic.

```
JAVA_OPTIONS=-verbose:gc
```

In WebSphere, specify this parameter in the administration console. In WebLogic, set this parameter in the start command file.

If you do not find the JAVA\_OPTIONS parameter, you can set the -verbose:gc parameter before the java command. For example, java -verbose:gc.

After you set garbage collection to verbose, the console or the log file displays how much time is spent for each garbage collection cycle.

Use this option only if you are concerned about garbage collection taking too long. Do not use this option on a production server because it slows the server.

The following figure shows the messages that the verbose garbage collection setting returns:

### Garbage Collection Statistics

```
C:\WINNT\system32\cmd.exe - startweblogicdebug
[GC 7146K->5268K(32576K), 0.0044318 secs ]
[GC 7316K->5409K(32576K), 0.0056742 secs ]
[GC 7457K->5650K(32576K), 0.0053113 secs ]
[GC 7698K->5785K(32576K), 0.0049800 secs ]
[GC 7833K->6052K(32576K), 0.0056826 secs ]
[GC 8100K->6297K(32576K), 0.0064771 secs ]
[Full GC 7101K->6401K(32576K), 0.1591987 secs ]
[GC 8448K->6455K(32576K), 0.0044584 secs ]
[GC 8503K->6599K(32576K), 0.0056471 secs ]
[GC 8646K->6635K(32576K), 0.0041659 secs ]
[GC 8683K->6797K(32576K), 0.0046836 secs ]
[GC 8845K->6879K(32576K), 0.0028638 secs ]
[GC 8927K->7105K(32576K), 0.0039622 secs ]
[GC 9153K->7146K(32576K), 0.0028783 secs ]
[GC 9194K->7265K(32576K), 0.0027805 secs ]
[GC 9313K->7955K(32576K), 0.0075627 secs ]
[GC 10003K->8476K(32576K), 0.0071506 secs ]
[GC 10524K->8725K(32576K), 0.0057530 secs ]
[GC 10772K->8898K(32576K), 0.0040514 secs ]
[GC 10946K->9391K(32576K), 0.0051241 secs ]
[GC 11439K->9626K(32576K), 0.0042626 secs ]
[GC 11670K->9807K(32576K), 0.0033948 secs ]
[GC 11854K->10199K(32576K), 0.0055549 secs ]
[GC 12247K->10383K(32576K), 0.0060340 secs ]
```

A line that starts with "Full GC" indicates that a complete garbage collection cycle was run. A line that starts with "GC" indicates that a minor garbage collection cycle was run.

The value to the left of the arrow (->) indicates the memory (or number of live objects) before garbage collection. The value to the right of the arrow indicates the

memory (or number of live objects) after garbage collection. The value in angle brackets (<>) indicates the total memory allocated for the heap at the time.

## Enabling Debug Mode for Maximo Enterprise Adapter Integration Logging

By default, debug mode for integration logging is not turned on. Complete the following steps to enable DEBUG mode logging for integration:

- 1 Open the logging.properties file and change the following properties:

Original property setting	Revised property setting
# log4j.logger.maximo.integration=INFO	log4j.logger.maximo.integration=DEBUG
log4j.rootLogger=ERROR, A1	log4j.rootLogger=ERROR, A1, A2

The A2 parameter writes errors to the maximo.log file.

- 2 Rebuild the EAR files.
- 3 Restart the application server.

## Troubleshooting Performance Issues in Application Server Configuration

This section describes issues that can result in poor performance in the application server. It also outlines recommendations that can help you to resolve those problems.

### Monitoring the Issues

The following table outlines the most common issues and their causes:

**Performance Problems**

Problem	Cause
Out of memory	<ul style="list-style-type: none"> <li>▼ Many users are directed toward a single instance of the application server.</li> <li>▼ A single instance of the application server is performing an excessive amount of work.</li> <li>▼ Heap memory is set too small.</li> <li>▼ Operations are fetching too much data at once from the database.</li> <li>▼ Bugs in the code hold onto objects in memory.</li> </ul>
Deadlocks	The Java code in the application or the Java Virtual Machine causes active threads to wait for resources that other threads are holding. This behavior eventually can cause all threads to wait.

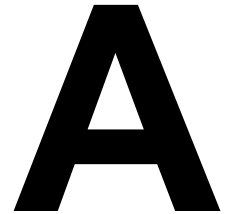
Study the log files of your application server to determine the garbage collection output, the heap dumps, and thread dumps.

**Addressing the Issues**

- ▼ Monitor the user load and server load. Use clustering and load balancing techniques to increase the number of server instances as necessary.
- ▼ Use 1 GB to 1.5 GB of heap memory.
- ▼ Use separate clusters for the user load system and background systems, such as cron tasks and integration module data loading.



# Setting Up Two Clusters



This appendix provides information about how to set up two clusters in an environment in which the Maximo Enterprise Adapter is deployed.

## Overview

Every external system that Maximo communicates with should be set up with its own sequential inbound and sequential outbound queues. If you also set up each sequential queue with its own cron task, administering the queues is easier.

For example, you might set up a system with three clusters, and call them UI, Q, and Cron.

- ▼ UI cluster: provides connection for Maximo application end users. Hosts the Web application and sequential outbound queues.
- ▼ Q cluster: principally for hosting continuous and sequential inbound queues.
- ▼ Cron cluster: where other Maximo cron tasks are configured.

The UI cluster is where users generate transactions when they save work in Maximo. The UI cluster should therefore host sequential outbound queues to send out user-generated Maximo transactions to external systems.

The purpose of clustering is to separate the inbound queue to a separate cluster from the interactive users. Separating the queue this way prevents inbound traffic from affecting the user interface performance.

The optional Cron cluster is set up for other Maximo cron tasks. Setting up a third cluster for cron tasks is helpful and should bring the system configuration close to its ideally segregated potential.

Cron cluster setup details are not described here or in the accompanying figures. If cron tasks might generate outbound Maximo transactions, then the Cron cluster needs its own set of sequential outbound queues. Outbound queue setup on the Cron cluster would be similar to outbound queue setup on the UI cluster and the Q cluster.

Maximo cron task configuration is detailed in the *Maximo System Administrator's Guide*.

# Example: Setting Up the UI Cluster and the Q Cluster

Following is a summary list of steps that you can use to set up the UI cluster and the Q cluster discussed in the previous section. The steps assume that there are two clusters. The UI cluster consists of eight JVMs. The Q cluster consists of two JVMs.

The naming convention uses the queue name appended with `_UI` or `_Q`, for example, `sqoutbd_ui` and `sqinbd_q`. Because each external system should have its own outbound queue and inbound queue, there are multiple instances of OUT and IN queues, for example, `SAP_SQOUT_UI`, `ORA_SQIN_Q`, and so on. (The example assumes that WebSphere is the application server.)

## Steps to Set Up the UI Cluster and the Q Cluster

- 1** Create two Maximo EAR files to deploy the Maximo applications on the two clusters.
  - a** For the UI cluster, edit the `deployment-application.xml` file to comment out `MEAWEB` and `MEAJMSEJB`.
  - b** For the Q cluster, edit the `deployment-application.xml` file to comment out `MaximoUIWEB`.
- 2** On WebSphere, create two database schemas.
- 3** Create eight JVMs for the UI cluster.
- 4** Create two JVMs for the Q cluster.
- 5** Create two clusters, UI and Q.
- 6** Add the eight JVMs that you created in step 3 to the UI cluster.
- 7** Add the two JVMs that you created in step 4 to the Q cluster.
- 8** Deploy two Maximo applications, one to the UI cluster, one to the Q cluster.
- 9** Create two JMS buses for the UI and Q clusters, for example, `meajmsbus_ui`, and `meajmsbus_q`.
- 10** Create two JDBC providers.
- 11** Create two data sources.
- 12** Create two authorization aliases that point to two schemas.
- 13** Add each cluster as a member to the respective bus. Ensure that you provide the correct data source JNDI names for each bus.
- 14** Add three destinations to the Q bus, for example:
  - `sqoutbd_q`

- sqinbd\_q
- cqinbd\_q

**15** Add a single destination to the UI bus, for example:

- sqoutbd\_ui

**16** Add these JMS objects to the UI cluster:

- One queue, and associate it with its respective bus and destination.  
For example, associate sqout\_ui with meajmsbus\_ui and sqoutbd\_ui.
- One connection factory and associate it with its respective bus.  
For example, associate conn\_fact\_ui with meajmsbus\_ui.

**17** Add these JMS objects to the Q cluster:

- Three queues, and associate them with their respective buses and destinations.  
For example, associate sqout\_q with meajmsbus\_q and sqoutbd\_q.  
Associate sqin\_q with meajmsbus\_q and sqinbd\_q.  
Associate cqin\_q with meajmsbus\_q and cqinbd\_q.
- One connection factory, and associate it with its respective bus.  
For example, associate conn\_fact\_q with meajmsbus\_q.
- One activation spec, and associate it with its respective bus and continuous queue JNDI name.  
For example, associate meajmsact with meajmsbus\_q and jms/mro/int/queues/cqin.

For queues and connection factories, use JNDI names that are based on the naming convention shown in the steps (for example, xxx\_ui and yyy\_q). Also be sure that these names match on the external system setup of the Maximo Enterprise Adapter.

If you model your clustering architecture on the example setup, be sure that JNDI names for connection factories and outbound queues are the same in the UI cluster and the Q cluster.

## Result of Setup

The result of the previous procedure is the creation of two messaging engines, one for each bus-cluster combination. Each engine has its own schema (pointed to by the data source of its respective cluster). One schema is for outbound sequential queue traffic of the UI cluster. The other schema is for inbound (sequential and continuous) and outbound (sequential) traffic of the Q cluster.

## Accessing the Sequential Queue

A potential issue in the example setup can occur if the UI cluster needs to access the sequential queue. This issue might happen if a user tries to import a file for data loading.

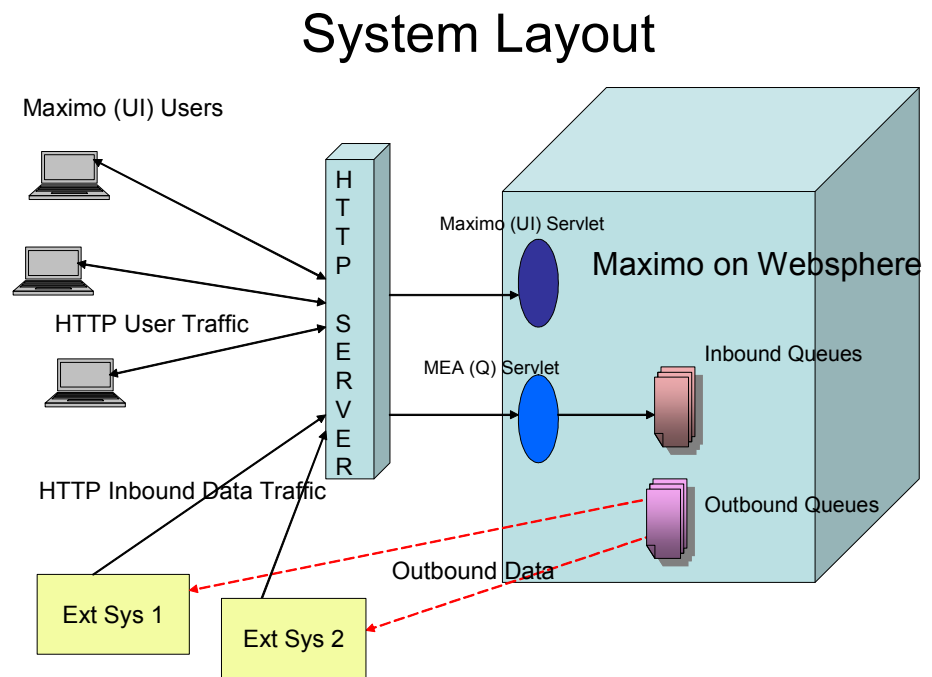
In such a situation, because there is no sequential queue visible at the UI cluster level, an error results. You can resolve the error by allowing the user to log in to the Q cluster Maximo application to perform the import operation. To allow the user to log in, open the respective port number on the virtual host setup.

An alternative is to raise the visibility level of a specific sequential queue so that it is visible at the UI cluster level and at the Q cluster level.

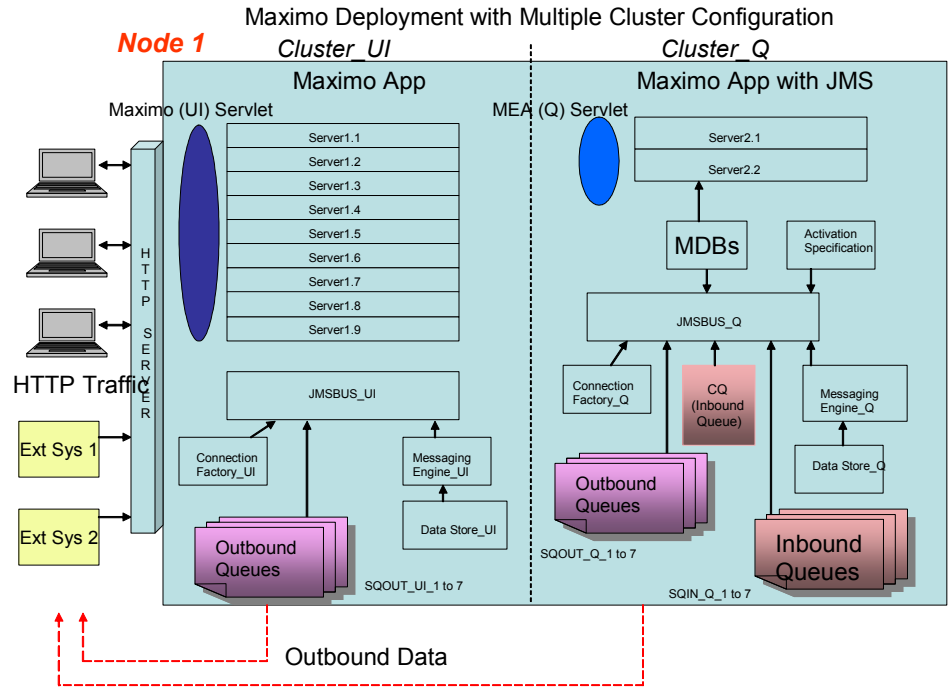
## Example Deployment and Configuration

The following figures illustrate the example deployment and configuration.

### System Layout



**Multiple Cluster Configuration**



In this setup, you might need to tune several settings. The numbers that are listed here are estimates. You might need to revise the values to optimize performance.

- ▼ Number of message-driven beans: 5
- ▼ Batch size: 5 connections
- ▼ JMS connection factory pool size: 50
- ▼ Default thread pool size: 50
- ▼ Memory allocation: 1.792 GB
- ▼ Cron task frequency for JMS queues: 30 seconds

## Example: Setting Up the UI Cluster and the Q Cluster

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
2Z4A/101  
11400 Burnet Road  
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

# Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM  
AIX  
DB2  
developerWorks  
Everyplace  
ibm.com  
Lotus  
Maximo  
Notes  
QuickPlace  
Tivoli  
WebSphere

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

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

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Pentium, and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

