Oracle® Enterprise Manager

System Monitoring Plug-in Metric Reference Manual for
Non-Oracle Middleware Management

Release 12 (12.0)

E10115-06

November 2010

Oracle Enterprise Manager System Monitoring Plug-in Metric Reference Manual for Non-Oracle Middleware Management, Release 12 (12.0)

E10115-06

Contents

3 IBM WebSphere MQ Metrics

4 JBoss Application Server Metrics

8 Microsoft Commerce Server Metrics

9 Microsoft Exchange Server Metrics

10 Microsoft Internet Information Services Metrics

11 Microsoft Internet Security and Acceleration Metrics

Preface

This manual is a compilation of the plug-ins metrics provided in Oracle Enterprise Manager for non-Oracle middleware management.

Audience

This document is intended for Oracle Enterprise Manager users interested in plug-ins metrics for database management.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documents

For more information, see the following documents in the Oracle Enterprise Manager 10*g* Release 2 documentation set:

- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Microsoft .NET Framework*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Microsoft BizTalk Server*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Microsoft Commerce Server*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Microsoft Active Directory*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Microsoft Internet Information Services*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Microsoft Internet Security and Acceleration Server*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

How to Use This Manual

The *System Monitoring Plug-in Metric Reference Manual for Non-Oracle Database Management* lists all the plug-ins metrics for database management that Enterprise Manager monitors. This manual shows all the metric help available online, eliminating the need to have the Grid Control Console up and running.

This preface describes:

- Structure of the Metric Reference Manual
- Background Information on Metrics, Thresholds, and Alerts

Structure of the Metric Reference Manual

This manual contains a chapter for each Enterprise Manager non-Oracle middleware management plug-in.

Metric Information

The information for each metric comprises a description and user action if available:

- Description

  Provides an explanation following the metric name. This text defines the metric and, when available, provides additional information pertinent to the metric.

- User Action (if available)

  Suggestions of how to solve the problem causing the alert.

Additional Information

The following table describes the additional information that may be provided for a metric.

Column Header	Column Definition
Target Version	Version of the target, for example, 9.0.2.x and 10.1.0.x. The x at the end of a version (for example, 9.0.2.x) represents the subsequent patchsets associated with that release.
Evaluation and Collection Frequency	The rate at which the metric is collected and evaluated to determine whether it has crossed its threshold. The evaluation frequency is the same as the collection frequency.

Column Header	Column Definition
Server Evaluation Frequency	The rate at which the metric is evaluated to determine whether it has crossed its threshold. For server-generated alerts, the evaluation frequency is determined by Oracle Database internals. For example, if the evaluation frequency is 10 minutes, then when the Average File Write Time degrades to the point an alert should trigger, it could be almost 10 minutes before Enterprise Manager receives indication of the alert. This column is present in the Metric Collection Summary table only for Oracle Database 10*g* metrics.
Collection Frequency	The rate at which the Management Agent collects data. The collection frequency for a metric comes from the Enterprise Manager default collection file for that target type.
Upload Frequency	The rate at which the Management Agent moves data to the Management Repository. For example, upload every n^{th} collection. The upload frequency for a metric comes from the Enterprise Manager default collection file for that target type. This column is present in the Metric Collection Summary table only when the Upload Frequency is different from the Collection Frequency.
Comparison Operator	The comparison method Enterprise Manager uses to evaluate the metric value against the threshold values.
Default Warning Threshold	Value that indicates whether a warning alert should be initiated. If the evaluation of the warning threshold value returns a result of TRUE for the specified number of consecutive occurrences defined for the metric, an alert triggers at the warning severity level.
Default Critical Threshold	Value that indicates whether a critical alert should be initiated. If the evaluation of the critical threshold value returns a result of TRUE for the specified number of consecutive occurrences defined for the metric, an alert triggers at the critical severity level.
Consecutive Number of Occurrences Preceding Notification	Consecutive number of times a metric's value reaches either the warning threshold or critical threshold before a notification is sent.
Alert Text	Message indicating why the alert was generated. Words that display between percent signs (%) denote variables. For example, `Disk Utilization for %keyValue% is %value%%` could translate to `Disk Utilization for d0 is 80%`.

Abbreviations and Acronyms

To reduce the page count in this document, the following abbreviations and acronyms are used:

Abbreviation/Acronym	Name
Agent	Oracle Management Agent
Database	Oracle Database
OMS	Oracle Management Service
Repository	Oracle Management Repository

Background Information on Metrics, Thresholds, and Alerts

A metric is a unit of measurement used to determine the health of a target. It is through the use of metrics and associated thresholds that Enterprise Manager sends out alerts notifying you of problems with the target.

Thresholds are boundary values against which monitored metric values are compared. For example, for each disk device associated with the Disk Utilization (%) metric, you can define a different warning and critical threshold. Some of the thresholds are predefined by Oracle, others are not.

Once a threshold is reached, an alert is generated. An alert is an indicator signifying that a particular condition has been encountered and is triggered when one of the following conditions is true:

- A threshold is reached.
- An alert has been cleared.
- The availability of a monitored service changes. For example, the availability of an application server changes from up to down.
- A specific condition occurs. For example, an alert is triggered whenever an error message is written to a database alert log file.

Alerts are detected through a polling-based mechanism by checking for the monitored condition from a separate process at regular, predefined intervals.

> **See Also:** See the *Oracle Enterprise Manager Concepts* manual and the Enterprise Manager online help for additional information about metrics, thresholds, and alerts

Editing

Out of the box, Enterprise Manager comes with thresholds for critical metrics. Warning and critical thresholds are used to generate an alert, letting you know of impending problems so that you can address them in a timely manner.

To better suit the monitoring needs of your organization, you can edit the thresholds provided by Enterprise Manager and define new thresholds. When defining thresholds, the key is to choose acceptable values to avoid unnecessary alerts, while still being notified of issues in a timely manner.

You can establish thresholds that will provide pertinent information in a timely manner by defining metric baselines that reflect how your system runs for a normal period of time.

The metrics listed on the Edit Thresholds page are either default metrics provided by Oracle or metrics whose thresholds you can change.

Specifying Multiple Thresholds

The Specifying Multiple Thresholds functionality allows you to define various subsets of data that can have different thresholds. By specifying multiple thresholds, you can refine the data used to trigger alerts, which are one of the key benefits of using Enterprise Manager.

The key in specifying multiple thresholds is to determine how the comparison relates to the metric threshold as a whole. What benefit will be realized by defining a more stringent or lax threshold for that particular device, mount point, and so on?

For example, using the Average Disk I/O Service Time metric, you can define warning and critical thresholds to be applied to all disks (sd0 and sd1), or you can define different warning and critical thresholds for a specific disk (sd0). This allows you to adjust the thresholds for sd0 to be more stringent or lax for that particular disk.

Accessing Metrics Using the Grid Control Console

To access metrics in the Grid Control Console, use the All Metrics page associated with a particular target by doing the following:

1. From the Grid Control Console, choose the target.
2. On the target’s home page, click All Metrics in the Related Links section.

3. On the All Metrics page, choose the metric of interest and click Help. The help for that metric displays.

1 Apache Tomcat Metrics

Oracle Enterprise Manager can be used to manage Apache Tomcat. You can use the **All Metrics** page for an Apache Tomcat target to view the metrics that are being monitored for that target by Oracle Management Agent.

1.1 Application Cache Metrics

This metric category contains the usage details of the application cache of the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–1 Application Cache Metrics

Metric	Description
Cache Access Count	Number of times the cache was accessed since last collection
Cache Hit Count	Number of times the cache was hit since last collection

1.2 All Applications Metrics

This metric category contains aggregated performance metrics spanning all the applications deloyed to the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–2 All Applications Metrics

Metric	Description
All Applications - Rejected Sessions	Number of sessions rejected by all the applications running on the Apache Tomcat container since last collection
All Applications - Sessions Created	Number of sessions created by all the applications running on the Apache Tomcat container since last collection
All Applications - Total Active Sessions	Active sessions of all the applications running on the Apache Tomcat container

1.3 All Applications JSP Monitor Metrics

This metric category contains aggregated performance metrics spanning all the JSPs deployed to the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–3 All Applications JSP Monitor Metrics

Metric	Description
All Applications - JSP Access	Number of times JSPs were accessed on all the applications since last collection
All Applications - JSP Reload	Number of times JSPs were reloaded on all the applications since last collection

1.4 All Global Request Processors Metrics

This metric category contains aggregated performance metrics spanning all the Global Request Processors deployed to the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–4 All Global Request Processors Metrics

Metric	Description
All Global Request Processors - Total Bytes Received	Bytes received by all the request processors running on the Apache Tomcat container since last collection
All Global Request Processors - Total Errors	Error count on all the request processors running on the Apache Tomcat container since last collection
All Global Request Processors - Total Requests	Request count on all the request processors running on the Apache Tomcat container since last collection
All Global Request Processors -Total Bytes Sent	Bytes sent by all the request processors running on the Apache Tomcat container since last collection

1.5 Applications Metrics

This metric category contains the metrics for each application deployed to the container.

Default Collection Interval — Every 5 minutes

Table 1–5 Applications Metrics

Metric	Description
Total Active Sessions	Total number of active sessions since startup
Total Rejected Sessions	Total number of rejected sessions since startup
Total Sessions Since Startup	Total number of sessions created since startup
Total Processing Time (ms)	Total processing time (in milliseconds) since startup
Total Request Count	Total number of requests served since startup
Processing Time Since Last collection (ms)	Processing time (in milliseconds) since last collection
Active Session Since last collection	Number of active sessions created since last collection
Sessions Created Since Last collection	Number of new sessions created since last collection
Rejected Sessions Since Last collection	Number of sessions rejected since last collection
Request Count Since Last collection	Number of requests served since last collection

Table 1–5 (Cont.) Applications Metrics

Metric	Description
Average Processing Time (ms)	Average time taken (in milliseconds) for processing a request

1.6 DataSource Metrics

This metric category contains the attributes of all the data sources configured for the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–6 DataSource Metrics

Metric	Description
Active Datasource Connections	Number of active connections to the data source
Idle Datasource Connections	Number of idle connections to the data source

1.7 Global Request Processor Metrics

This metric category contains the attributes of each Global Request Processor configured on the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–7 Global Request Processor Metrics

Metric	Description
Average Request Processing Time (ms)	Average time taken (in milliseconds) for processing a request by the global request processor
Bytes Received Since Last Collection	Bytes received by each request processor since last collection
Bytes Sent Since Last Collection	Bytes sent by each request processor since last collection
Errors Since Last Collection	Error count of each request processor since last collection
Request Processing Time (ms)	Request processing time (in milliseconds) of each global request processor
Requests Since Last Collection	Number of requests served by each request processor since last collection

1.8 JSP Monitor Metrics

This metric category contains the usage details of JSPs.

Default Collection Interval — Every 5 Minutes

Table 1–8 JSP Monitor Metrics

Metric	Description
JSP Access	Number of times JSPs were accessed since during last collection
JSP Reload	Number of times JSPs were reloaded since last collection

1.9 Memory Usage Metrics

This metric category gives information about the memory load on the Apache Tomcat container.

Default Collection Interval — Every 5 minutes

Table 1–9 Memory Usage Metrics

Metric	Description
Heap Memory Usage (MB)	Heap memory usage (in MB) of the server
Non Heap Memory Usage (MB)	Non-heap memory usage (in MB) of the server

1.10 Response Metrics

This metric category shows the status of the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–10 Response Metrics

Metric	Description
Status	Status of the Apache Tomcat container - whether up or down

1.11 Servlets Metrics

This metric category contains attributes of each servlet loaded on the Apache Tomcat container.

Default Collection Interval — Every 5 minutes

Table 1–11 Servlets Metrics

Metric	Description
Maximum Servlet Request Processing Time (ms)	Maximum time taken (in milliseconds) for processing a request by the servlet
Total Servlet Request Processing Time (ms)	Total time taken (in milliseconds) for processing all the requests by the servlet
Number Of Requests Since Startup	Number of requests received by the servlet since startup
Number Of Errors Since Startup	Number of erroneous requests received by the servlet since startup
Processing Time Since Last Collection	Time taken for processing all the requests since last collection
Requests Count Since Last Collection	Number of requests served by the servlet since last collection
Average Processing Time (ms)	Average processing time (in milliseconds) since last collection
Number Of Servlet Errors Since Last Collection	Number of erroneous requests received by the servlet since last collection

1.12 Session Manager Metrics

This metric category contains the statistics of the sessions managed by the Apache Tomcat container.

Default Collection Interval — Real-Time Only

Table 1–12 Session Manager Metrics

Metric	Description
Active Sessions Since Startup	Active number of sessions of each application since startup
Rejected Sessions Since Startup	Number of sessions rejected by each application since startup
Application Name	Name of the deployed application
Sessions Created Since Startup	Number of new sessions created since startup
Sessions Created Since Last Collection	Number of sessions created since last collection
Active Sessions Since Last Collection	Number of active sessions since last collection
Rejected Sessions Since Last Collection	Number of sessions rejected since last collection

1.13 Thread Pools Metrics

This metric category contains statistics on various thread pools used by the Apache Tomcat container.

Default Collection Interval — Every 5 Minutes

Table 1–13 Thread Pools Metrics

Metric	Description
Busy Threads	Number of busy threads on the Apache Tomcat container
Threads	Number of threads created on the Apache Tomcat container

2

IBM WebSphere Application Server Metrics

IBM WebSphere Application Server is one of the non-Oracle application server that is monitored using Oracle Enterprise Manager Grid Control.

This chapter provides descriptions for all IBM WebSphere Application Server metric categories, and tables list and describe associated metrics for each category.

2.1 Applications Metrics

This category of metrics describes the applications running on the server.

Default Collection Interval — Every 15 minutes

Table 2–1 Applications Metrics

Metric	Description
Active Requests	Current active requests of the application
Average EJB Method Execution Time	Average EJB method execution time
No. of Active EJB Methods	Number of active EJB methods
Request Processing Time	When a request comes to an application server, it passes through a servlet and is then processed. This metrics depicts the time taken by the server to fulfill this request. Since the servlet is the entry or exit point of the request, this also describes the time taken by the servlet to fulfill a request

2.2 Dynamic Cache Metrics

This category of metrics provides information about the dynamic cache.

Default Collection Interval — Every 15 minutes

Table 2–2 Dynamic Cache Metrics

Metric	Description
InMemory Cache Size	Size of the InMemory cache
Max InMemory Cache Size	Maximum memory that can be used by the cache
Total Timeout Invalidations	Total number of timeout invalidations that have occurred at the server

2.3 EJB Metrics

This category of metrics provides information about the EJBs that were received and responded to maximum requests.

Default Collection Interval — Every 15 minutes

Table 2–3 EJBModule Metrics

Metric	Description
Active Methods	Number of methods of the EJBs that are currently active
Current Average Execution Time	Average method execution time for the current period
Method Execution Rate per second	Indicates the EJB method execution rate per second
Total Method Calls	Total method calls on the EJBs

2.4 EJBModule Metrics

This category of metrics describes the EJB module deployed on the server.

Default Collection Interval — Every 15 minutes

Table 2–4 EJBModule Metrics

Metric	Description
Active Methods	Number of methods of the EJBs that are currently active
Average method execution time during last interval	Average execution time of the method during the last interval
Avg Method Response Time	Average response time of the method
Concurrent Lives	Concurrent lives
Current average method execution time	Current average method execution time
Destroys	Number of EJBs that were destroyed by the container
Instantiates	Number of new instances made
Number of EJB Methods Executed during last interval	Number of EJB methods executed during the last interval
Ready Count	Number of EJBs in the ready state
Total Method Calls	Total method calls on the EJBs

2.5 EntityBean Metrics

This category of metrics describes the EJB module deployed on the server.

Default Collection Interval — Every 15 minutes

Table 2–5 Entity Bean Metrics

Metric	Description
Activates	Number of times an instanced was activated
Activation Time	Time taken by the EJB container to activate the EJB instance
Active Methods	Number of methods which can be invoked for action simultaneously
Average method execution time during last interval	Average method execution time during the last interva
Avg Drain Size	Average size of the drain

Table 2–5 (Cont.) Entity Bean Metrics

Metric	Description
Concurrent Lives	Number of EJB beans in the live state in the container at the same time. In the live state, the client can invoke a method on these beans for business processes.
Creates	Number of new instances that were created for the EJB
Current average execution time	Number of EJB methods that were executed during the last interval
Destroys	Number of instances that were destroyed
Drains From Pool	Number of instance drains from the pool
EJB Method Execution Rate per second	EJB method execution rate per second
Gets Found	Number of Gets found
Gets From Pool	Number of times the instance was taken from the pool
Instantiates	Number of new instantiates of the EJB object
Load Time	Time taken for loading beans in the server
Loads	Number of times an instance was loaded
No of EJB Methods Executed during last interval	Number of EJB methods that were executed during the last interval
Passivates	Number of times this EJB instance was passivated
Passivation Time	Time taken by the container to passivate the bean
Pool Size	Size of the EJB instances pool
Ready Count	Number of instances that are in the ready state
Removes	Number of instances of this EJB that were removed from the container
Returns Discarded	Number of returns that were discarded
Returns To Pool	Number of instances that were returned to the pool after usage
Store Time	Store time for the EJB instance
Stores	Entity beans are stored in persistent storage devices such as relational databases so that they can be recreated even after the container goes down. This metric depicts the number of times the entity beans instance state is stored in the database or persistent storage, instead of being stored in the container.

2.6 J2EE Server Metrics

This category of metrics describes the current server statistics.

Default Collection Interval — Every 15 minutes

Table 2–6 J2EE Server Metrics

Metric	Description
Active EJB Methods	Number of active methods
Active Requests	Number of active requests
Active Sessions	Number of active sessions
Active Transactions	Number of active transactions
Delta Method Call	Number of EJB methods that were executed during the last interval

Table 2–6 (Cont.) J2EE Server Metrics

Metric	Description
EJB Method Execution Rate per second	EJB method execution rate per second
EJB Method Execution Time	Time taken by EJBs for method execution
Live Sessions	Number of live sessions (including those that are alive but not active.)
Open JDBC Connections	Number of open JDBC connections
Request Processing Time	Time taken to process requests
Requests Processed for second	Number of requests processed per second

2.7 JDBCProvider Metrics

This category of metrics depicts the name of the provider software that is being used to connect to a relational database using JDBC.

Default Collection Interval — Every 15 minutes

Table 2–7 JDBC Provider Metrics

Metric	Description
Average Wait Time	Average waiting time for a new connection
Avg Use Time	Average time a provider was used
Concurrent Waiters	Number of connections that are in the *waiting* state
Faults	Number of faults
Free Pool Size	Size of the free pool
JDBC Operation Timer	JDBC operation timer
Number Allocates	Number of allocations
Number Connection Handles	Number of connections that are currently being handled
Number Creates	Number of times a new instance of the provider was created
Number Destroys	Number of times the instance of this provider was destroyed
Number of Managed Connections	Number of managed connections
Number Returns	Number of returns
Percent Maxed	Percentage maxed
Percent Used	Percentage used
Pool Size	Size of the pool
Prepared Statement Cache Discards	Count of prepared statement cache discard and the statement recompiled

2.8 JSP Metrics

This category of metrics provides information about the JSPs running on this server.

Default Collection Interval — Every 15 minutes

Table 2–8 JSP Metrics

Metric	Description
Active Requests	Number of requests that are currently being processed by the JSP
Average Processing Time (seconds)	Average JSP processing time
Concurrent Requests	Number of concurrent requests
Current Client Processing Time	Current client processing time
Number Errors	Number of errors that were generated while displaying the JSP
Requests Per Second During Last Interval	Number of active requests per second during the last interval
Response Time	JSP response time in seconds
Time Taken During Last Interval	Time taken during the last interval
Total Requests	Total number of requests received

2.9 JVM Metrics

This category of metrics provides information about the Java Virtual Machines.

Default Collection Interval — Every 15 minutes

Table 2–9 JVM Metrics

Metric	Description
CPU Utilization (%)	
Free Memory	Amount of free memory available
Total Memory	Total memory used by the server
Up Time	Time for startup of the server after activation
Used Memory	Memory used by the server

2.10 MessageDrivenBean Metrics

This category of metrics provides information about the message-driven bean.

Default Collection Interval — Every 15 minutes

Table 2–10 MessageDrivenBean Metrics

Metric	Description
Active Methods	Number of active methods
Avg Drain Size	Average drain size
Avg Server Session Wait Time	Average server session wait time
Avg method exec. time during last interval	Average method execution time during the last interval
Concurrent Lives	Number of concurrent lives
Current avg exec. time	Current method execution time
Delta Method Calls	Number of EJB methods that were executed during the last interval
Destroys	Number of instances that were destroyed
Drains From Pool	Number of drains from the pool of EJB instances

Table 2–10 (Cont.) MessageDrivenBean Metrics

Metric	Description
EJB Method Execution Rate per second	EJB method execution rate per second
Gets Found	Number of gets that were found
Gets From Pool	Number of gets from the pool of EJB instances
Instantiates	Number of new instances
Message Blackout Count	Number of messages that were blacked out
Message Count	Number of messages
Pool Size	Size of the pool
Returns Discarded	Number of returns that were discarded by the pool
Returns To Pool	Number of returns to the pool
Server Session Usage	Usage of the server session

2.11 ORB Metrics

This category of metrics describes the Object Request Broker (ORB).

Default Collection Interval — Every 15 minutes

Table 2–11 ORB Metrics

Metric	Description
Concurrent Requests	Number of concurrent requests
Reference Lookup Time	Time taken to look up a reference of the Object Request Broker
Total Requests	Total number of requests to the Object Request Broker

2.12 Response

This category of metrics provides basic information about the IBM WebSphere Application Server.

2.12.1 Status

This metric indicates whether the IBM WebSphere Application Server is up or down. If the value is 1, the server is up. If the value is 0, it is down.

2.12.1.1 Metric Summary

The following table shows how often the metric's value is collected and compared against the default thresholds. The 'Consecutive Number of Occurrences Preceding Notification' column indicates the consecutive number of times the comparison against thresholds should hold TRUE before an alert is generated.

Table 2–12 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every Minute	After Every 60 Samples	=	Not Defined	0	1	The J2EE Server instance is down

2.13 Servlet Metrics

This category of metrics provides information about the servlets running on this IBM WebSphere Application Server.

Default Collection Interval — Every 15 minutes

Table 2–13 Servlet Metrics

Metric	Definition
Active Requests	Number of requests that are currently being processed by the servlet
Average Processing Time (seconds)	
Concurrent Requests	Number of concurrent requests sent to the servlet
Current Client Processing Time	Current client processing time in seconds
Number Errors	Number of errors that were generated while responding to a request
Requests Per Second During Last Interval	Number of active requests per second during the last interval
Response Time	Response time in seconds
Time Taken During Last Interval	Time take during the last interval
Total Requests	Total number of requests received by the servlet

2.14 SessionManager Metrics

This category of metrics describes the session managers created for web applications.

Default Collection Interval — Every 15 minutes

Table 2–14 SessionManager Metrics

Metric	Description
Activate NonExist Sessions	Number of nonexistent sessions that are activated
Active Sessions	Number of active sessions
Affinity Breaks	Number of session affinity breaks
Cache Discards	Number of times that the cache was discarded
Created Sessions	Number of session objects that were created by the server
External ReadSize	External read size
External ReadTime	External read time
External Write Size	External write size for this session

Table 2–14 (Cont.) SessionManager Metrics

Metric	Description
External WriteTime	External write time
Invalidated Sessions	Number of sessions that were invalidated
Invalidated Via Timeout	Number of sessions that were invalidated by timeouts
Live Sessions	Number of live sessions till date
No Room For New Session	No room for new sessions
Session Life Time	Session's life time
Time Since Last Activated	Time since this session was last activated

2.15 Stateful SessionBean Metrics

This category of metrics provides information about the stateful session EJB deployed on the IBM WebSphere Application Server.

Default Collection Interval — Every 15 minutes

Table 2–15 Stateful SessionBean Metrics

Metric	Description
Average Method Execution Time	Average response time of the methods that were executed during the last interval
Activates	Number of time the EJB instance was activated by the container
Activation Time	Time taken by the container to put the EJB instance in an active state
Active Methods	Number of methods that are currently being invoked by the EJB
Concurrent Lives	Number of concurrent lives
Creates	Number of times a new instance of this EJB was created by the container
Current Average Execution Time	Current average execution time of methods by the EJB
Destroys	Number of times the EJB instance was destroyed by the container
EJB Method Execution Rate per second	Rate of execution of EJB methods per second
EJB Method Call Count during Last Interval	Number of EJB methods that were executed during the last interval
Instantiates	Number of times the EJB was instantiated after it was destroyed
Passivates	Number of times the EJB was passivated from the EJB container
Passivation Time	Time taken by the EJB container to change the state of the EJB from active to passivation
Ready Count	Number of EJBs that are in the *ready* state in the containers
Removes	Number of times the EJB was removed from the EJB container

2.16 StatelessSessionBean Metrics

This category of metrics provides information about the stateless session bean.

Default Collection Interval — Every 15 minutes

Table 2–16 StatelessSessionBean Metrics

Metric	Description
Active Methods	Number of methods that are currently active
Avg Drain Size	All EJB objects exist in a container. The container can discard them by either destroying or passivating them. This metric indicates the drain size, that is the number of objects removed from a container at a time, or in one drain.
Average Method Execution Time	Average method execution time during the last interval
Concurrent Lives	Number of concurrent lives
Current Average Execution Time	Current method execution time
Delta Method Calls	Number of EJB methods that were executed during the last interval
Destroys	Number of instances that were destroyed by the container
Drains From Pool	Number of drains from the pool
EJB Method Execution Rate per second	EJB method execution rate per second
Gets Found	Number of gets that were found
Gets From Pool	Number of times instances were fetched from the pool
Instantiates	Number of new EJB instances
Pool Size	Size of the pool
Ready Count	Number of instances that are in the ready state
Returns Discarded	For a stateless or entity bean, after an object is utilized for some action or business process, it is sent back to the container so that it can be reused. However, when the object is in process, the container may recreate more objects in the pool so that when this particular object is returned to the container the container may not be able to accept it because its pool size is full. Hence these objects are discarded. This metrics depicts the number of such objects.
Returns To Pool	Number of returns to the pool

2.17 System Metrics

This category of metrics provides information about the system that is hosting this IBM WebSphere Application Server.

Default Collection Interval — Every 15 minutes

Table 2–17 System Metrics

Metric	Description
Avg CPU Utilization	Average utilization of the system’s CPU
CPU Utilization	Current utilization of the system’s CPU
Free Memory	Amount of free memory currently available in the system

2.18 ThreadPool Metrics

This category of metrics provides information about the thread pool that contains the current running thread.

Default Collection Interval — Every 15 minutes

Table 2–18 ThreadPool Metrics

Metric	Description
Active Threads	Number of currently active threads
Percent Maxed	Maximum percentage of threads at a time in the pool
Pool Size	Maximum number of threads a pool can contain
Thread Creates	Number of threads that were created in this pool
Thread Destroys	Number of times a thread pool was destroyed

2.19 TransactionService Metrics

This category of metrics describes the transaction service. The following table lists the metrics and their associated descriptions.

Default Collection Interval — Every 15 minutes

Table 2–19 TransactionService Metrics

Metric	Description
Active Global Transactions	Number of global transactions that are active
Global Before Completion Duration	Global before completion duration
Global Commit Duration	Time taken for committing global transaction
Global Prepare Duration	Global prepare duration
Global Transaction Duration	Time taken by global transactions
Global Transaction Timeout	Number of transactions that were timed out
Global Transactions Begun	Number of global transactions that were initiated
Global Transactions Committed	Number of global transactions that were committed
Global Transactions Involved	Global transactions involved
Global Transactions RolledBack	Number of transactions that were rolled back
Local Before Completion Duration	Local before completion duration
Local Commit Duration	Time taken for committing a local transaction
Local Transaction Duration	Local transaction duration
Local Transaction Timeout	Number of local transactions that were stuck and timed out
Local Transactions Begun	Number of local transactions that have begun
Local Transactions Committed	Number of local transactions that were committed
Local Transactions RolledBack	Time taken for rolling back local transactions
Number Optimizations	Number of optimizations

2.20 WebModule Metrics

This category of metrics provides information about the web module or war file deployed on this server.

Default Collection Interval — Every 15 minutes

Table 2–20 WebModule Metrics

Metric	Description
Number Loaded Servlets	Number of servlets that were reloaded into the server
Number Reloads	Number of reloads of the web application inside the server memory

2.21 WebServices Service Metrics

This category of metrics provides information about the webservices running on this IBM WebSphere Application Server.

Default Collection Interval — Every 15 minutes

Table 2–21 WebServices Service Metrics

Metric	Description
Dispatch Response Time	Dispatch response time
Number Dispatched	Number of replies dispatched
Number Loaded Services	Number of loaded services
Number Received	Number of requests received by this webservice
Number Successful	Number successful
Reply Response Time	Time taken by the webservice to respond to a request
Reply Size	Size of the reply sent by this webservice
Request Response time	Request response time
Request Size	Size of the request
Response Time	Time taken to respond to a request
Size	Size

3

IBM WebSphere MQ Metrics

WebSphere MQ (formerly MQSeries) is IBM's Message Oriented Middleware offering that allows independent and potentially non-concurrent applications on a distributed system to communicate with each other.

This chapter provides descriptions for all IBM WebSphere MQ metric categories, and tables list and describe associated metrics for each category.

3.1 Channel Current Status Metrics

This category provides metrics about the current status of the channel.

Default Collection Interval — Every 3 minutes

Table 3–1 Channel Current Status Metrics

Metric	Description
Bytes Received	Shows the number of bytes received.
Bytes Sent	Shows the number of bytes sent
Delta Bytes Received	Shows the number of bytes received since last collection
Delta Bytes Sent	Shows the number of bytes sent since last collection
Delta Messages Sent/Received	Shows the number of messages processed since last collection
Messages Sent/Received	Shows the number of messages processed
Status	Shows the current status.

3.2 Channel Saved Status Metrics

This category provides metrics about Channel Saved Status.

Default Collection Interval — Every 2 minutes

Table 3–2 Channel Saved Status Metrics

Metric	Description
Last Seq Number	Shows the sequence number of the last message sent
Status	Shows the status of the channel
Type	Shows the type of the channel
Xmit Queue	Shows the transmission queue used by sender channels

3.3 Channels Metrics

This category provides metrics related to the channels.

Default Collection Interval — Every 10 minutes

Table 3–3 Channels Metrics

Metric	Description
Alteration Date	Shows the date on which the definition was last altered
Cluster Name	Shows the name of the IBM WebSphere MQ Cluster to which this channel belongs
Cluster Name list	Shows the cluster name list
Connection Name	Shows the name of the channel connection
Max Msg Length	Shows the maximum length of the message that can be processed by the channel
Type	Shows the type of the channel
Xmit QName	Shows the name of the transmission queue used by the sender channels

3.4 DynamicQs Metrics

This category provides metrics related to Dynamic Queues.

Table 3–4 DynamicQs Metrics

Metric	Description
Definition Type	Shows the type of queue definition
Type	Shows the type of the queue
Usage	Shows information about the usage (normal/transmission)

3.5 Local Queues Metrics

This category shows metrics related to local queues.

Default Collection Interval — Every 10 minutes

Table 3–5 Local Queues Metrics

Metric	Description
Cluster Name List	Shows the cluster Name List
Definition Type	Shows the type of the queue definition
Depth	Shows the number of messages on the queue
Max Depth	Shows the maximum number of messages on the queue
Max Message Length	Shows the maximum length of the message can be processed
Open Input Count	Shows the number of open input handles, that is the total handles open for input on this queue. This gives you an idean on the number of applications that are currently connected to the queue to put messages on the queue
Open Output Count	Shows the number of open output handles, that is the total handles open for output on this Queue. This gives you an idea on the number of applications that are currently connected to the queue to get messages from the queue
Percent Usage	Shows the space used (in percentage)

Table 3–5 (Cont.) Local Queues Metrics

Metric	Description
Type	Shows the type of the queue
Usage	Shows details about the usage, which can be "Normal" or "Trasmission")
Xmit QName	Shows the name of the transmission queue

3.6 QM Cluster Metrics

This category provides metrics related to IBM Websphere MQ Queue Manager Cluster, mainly the details about the channel used, the queue manager within this cluster, and the type of queue manager that is within this cluster.

Default Collection Interval — Every 10 minutes

Table 3–6 QM Cluster Metrics

Metric	Description
Queue Manager Name	Shows the name of the IBM Websphere MQ Queue Manager that is within this IBM Websphere MQ Queue Manager Cluster
Queue Manager Type	Shows the type of the IBM Websphere MQ Queue Manager (full/partial repository)

3.7 QM General Metrics

This caterogy provides metrics related to QM General.

Default Collection Interval — Every 2 minutes

Table 3–7 QM General Metrics

Metric	Description
Command QName	Shows the name of the local queue that is used for accepting commands
Dead Letter QName	Shows the name of the local queue that is to be used for undelivered messages
Default Xmit QName	Shows the name of the default transmission queue that is used for the transmitting messages to remote queue managers, if there is no other indication of which transmission queue to use
Max Handles	Shows the maximum number of handles that any one job can have open at a time
Max Message Length	Shows the maximum length of message that can be processed
Platform	Shows the platform on which the queue manager is running
Repository Name List	Shows a list of clusters for which this queue manager provides a repository service

3.8 QMLoad Metrics

This category provides metrics about the load on IBM Websphere MQ Queue Manager.

Table 3–8 QMLoad Metrics

Metric	Description
Bytes Received	Shows the bytes received since last collection

Table 3–8 (Cont.) QMLoad Metrics

Metric	Description
Bytes Sent	Shows the bytes sent since last collection
Messages Sent/Received	Shows the messages sent or received since last collection

3.9 Queue Status Input Handle Metrics

This category provides metrics related to Queue Status Input Handle.

Default Collection Interval — Every 10 minutes

Table 3–9 Queue Status Input Handle Metrics

Metric	Description
Appl Tag	Shows the name of the application that is using this queue
Appl Type	Shows the type of the application that is using this queue
Channel Name	Shows you the channel used for the message input
Connection Name	Shows the name of the connection used by the channel
Open Options	Shows the open options for the queue
User Identifier	Shows you the user ID of the application

3.10 Queue Status Output Handle Metrics

This category provides metrics related to Queue Status Output Handle.

Default Collection Interval — Every 10 minutes

Table 3–10 Queue Status Output Handle Metrics

Metric	Description
Appl Tag	Shows the name of the application that is using this queue
Appl Type	Shows the type of the application that is using this queue
Channel Name	Shows the name of the channel used for the message output
Connection Name	Shows the name of the connection name used by the channel
Open Options	Shows the open options for this queue
User Identifier	Shows the user ID of the application

3.11 Queues Metrics

This category provides metrics related to Message Queues.

Default Collection Interval — Every 10 minutes

Table 3–11 Queues Metrics

Metric	Description
Cluster Name List	Shows the list of clusters with which the queue is shared

Table 3–11 (Cont.) Queues Metrics

Metric	Description
Depth	Shows the current depth (number of messages) of the queue. For this metric you can set different warning and critical threshold values for each "QName" object. If warning or critical threshold values are currently set for any "QName" object, those thresholds can be viewed on the Metric Detail page for this metric.
Max Depth	Shows the maximum depth (number of messages) of the queue
Max Message Length	Shows the maximum message length allowed on a queue
Open Input Count	Shows the number of open input handles, that is the total handles open for input on this Queue. This gives you an idea on the number of applications that are currently connected to the queue to put messages on the queue
Open Output Count	Shows the number of open output handles, that is the total handles open for output on this Queue. This gives you an idea on the number of applications that are currently connected to the queue to get messages from the queue
Remote QManager	Shows the name of the remote queue manager (only for remote queues)
Remote QName	Shows the remote queues with which this queue interacts
Type	Shows the queue type. It can be one of the following: Local, Remote, Model, Alias Queue, or Read-only
Use	Shows how the queue is being used. The usage types can be "Normal" or "Tramission"
Xmit QName	Shows the transmission queue associated with this queue (only for remote queues)

3.12 Response Metrics

This category provides status metrics.

Table 3–12 Response Metrics

Metric	Description
Response	Shows the current status of the Queue Manager. The evaluation and collection frequency is 1 minute. The upload frequency is after every 5 samples.

3.13 Sender Channel Qs Metrics

This category provides metrics related to the Sender Channel Queues.

Default Collection Interval — Every 10 minutes

Table 3–13 Sender Channel Qs Metrics

Metric	Description
Xmit QName	Shows the name of the transmission queue

4

JBoss Application Server Metrics

JBoss Application Server is the market-leading, open source Java 2 Enterprise Edition (J2EE) application server, delivering a high-performance and enterprise-class platform for e-business applications. JBoss provides enterprise-class security, transaction support, resource management, load balancing, and clustering.

This chapter provides descriptions for all JBoss Application Server metric categories, and tables list and describe associated metrics for each category.

4.1 JBoss Clustering Service Metrics

This category provides metrics about High Availability JNDI (HAJNDI) service of JBoss.

Table 4–1 JBoss Clustering Service Metics

Metric	Description
Partition Name	Shows the name of the partition under which JBoss Application Server is running.
Service - Bind Address	Shows the IP address to which the High Availability JNDI (HAJNDI) Service is bound.
Service - Port	Shows the port used by the HAJNDI service.
Service - Rmi Port	Shows the RMI port of the HAJNDI service.
Service - State	Shows the state of the HAJNDI service. Number "3" is for "OK" state.
Service - State (Text)	Shows the state of the HAJNDI service.

4.2 JBoss Connection Pools Metrics

This category provides Managed Connection Pool details for JBoss Application Server.

Table 4–2 JBoss Connection Pools Metrics

Metric	Description
Criteria	Shows the criteria for managed connection pool for JBoss Application Server.
Maximum Size	Shows the maximum size for managed connection pool for JBoss Application Server.
Minimum Size	Shows the maximum size for managed connection pool for JBoss Application Server.

4.3 JBoss DataSource Bind Metrics

This category provides DataSourceBinding service details for JBoss Application Server.

Table 4–3 JBoss DataSource Bind Metrics

Metic	Description
JNDI Name	shows the JNDI Name for DataSourceBinding service for JBoss Application Server.

4.4 JBoss Ejb Modules Metrics

This category shows the information for the EJB modules that are loaded inside the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–4 JBoss Ejb Modules Metrics

Metric	Description
Name	Shows the name of the EJB module.
Stastistic Provider	This metric shows that the "Statistic Provider" and "State Manageable" attributes are provided to inform a client that the EJB module that is being looked at does or does not support state management, or does or does not provide statistics as defined by the JSR-77 specification.

4.5 JBoss Entity Ejb Metrics

This metric category shows information for all the entity EJBs loaded inside the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–5 JBoss Entity Ejb Metrics

Metric	Description
Entity EJB - Current Instances Pooled Count	Shows the number of instance kept in the pool currently.
Entity EJB - Current Ready Instances Count	Shows the number of instance of this EJB are in the ready state.
Entity EJB - Instances Created Count	Shows the count of the instances created for this EJB by the EJB container.
Entity EJB - Instances Removed Count	Shows the number of instance of this EJB that are removed from the EJB container.
Entity EJB - Maximum Instances in the Pool	Shows the maximum number of instance of this EJB that can be put in the pool.
Entity EJB - Maximum Ready Instances Count	shows the maximum number of instance of this EJB that can be in the passive state.
Entity EJB - Minimum Instances in the Pool	Shows the minimum number of instance of this EJB that should be there in the pool.
Entity EJB - Minimum Ready Instances Count	Shows the minimum number of instance of this EJB that should be in the ready state.
Entity EJB - Name	Shows the name of the EJB.
Entity EJB - Parent	Shows the parent or the JAR file to which this EJB belongs.

Table 4–5 (Cont.) JBoss Entity Ejb Metrics

Metric	Description
Statistic Provider	Shows performance statistics support for this managed object (whether true or false).

4.6 JBoss JCA Metrics

This metric category gives information for the Java Connector (JCA) resources of the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–6 JBoss JCA Metrics

Metric	Description
JCA - Count Use Time	Shows the current use time (in ms) for JCA Resource.
JCA - Count Use Time (Sec)	Shows the current use time for JCA Resource (in seconds).
JCA - Count Wait Time	Shows the current wait time (in ms) for JCA Resource.
JCA - Current Free Pool Size	Shows the number of instances of this resource in the pool that are not being used currently.
JCA - Current Pool Size	Shows number of instances in the pool currently.
JCA - Current Waiting Thread Count	Shows the number of threads waiting to acquire JCA connections.
JCA - High Pool Size	Shows the High Water mark for JCA Connection pool.
JCA - Instances Close Count	Shows the number of instances closed.
JCA - Instances Created Count	Shows the number of instances created.
JCA - Low Pool Size	Shows the Low Water mark for JCA Connection pool.
JCA - Lower Bound Free Pool Size	Shows the minimum number of instances that can be free inside the pool.
JCA - Lower Bound Pool Size	Shows the lower bound for lower connection pool.
JCA - Max Wait Time	Shows the minimum wait time (in ms) for JCA Resource.
JCA - Maximum Pool Size	Shows the maximum number of instances of this resource that can be there in the pool.
JCA - Maximum Use Time	Shows the maximum usage time for JCA Resource.
JCA - Maximum Waiting Thread Count	Shows the high water mark for waiting thread count for JCA connections.
JCA - Min Wait Time	Shows the minimum wait time (in ms) for JCA Resource.
JCA - Minimum Pool Size	Shows the minimum free pool size.
JCA - Minimum Used Time	Shows the minimum usage time for JCA Resource.
JCA - Minimum Waiting Thread Count	Shows the low water mark for waiting thread count for JCA connections.
JCA - Total Used Time	Shows the total usage time for JCA Resource.
JCA - Total Wait Time	Shows the total wait time for JCA Resource (in ms).
JCA - Upper Bound Free Pool Size	Shows the maximum number of instances that can be free inside the pool.
JCA - Upper Bound Pool Size	Shows the upper bound for JCA Connection Pool.
Name	Shows the name of the JCA resource.

Table 4–6 (Cont.) JBoss JCA Metrics

Metric	Description
Parent	Shows the name of the resource adaptor to which the JCA resource belongs.
Statistics Provider	Shows performance statistics support for this managed object (whether true or false).

4.7 JBoss JVM Metrics

This metric category provides information about the Java Virtual Machine (JVM) on which the JBoss Application Server is running. The upload frequency of this metric is after every sample and the Default Warning Threshold and Default Critical Threshold are 25 and 10 respectively.

Default Collection Interval — Every 30 minutes

Table 4–7 JBoss JVM Metrics

Metric	Description
JVM - Free Heap (%)	Shows the percentage of heap memory that is free and is not used by the JVM.
JVM - Heap Size (MB)	Shows the heap memory (in MB) used by the JVM that is used by JBoss Application Server.
JVM Name	Shows the name of the JVM on which JBoss Application Server is running.
JVM Version	Shows the version of the JVM that is used by the JBoss Application Server.
Vendor	Shows the name of the vendor or provider of the JVM.

4.8 JBoss Mail Service Metrics

This metric category provides information about the JBoss Mail Service.

Table 4–8 JBoss Mail Service Metrics

Metric	Description
Service - Default Sender	Shows the default sender of the any message that goes via JBoss mail service.
Service - JNDI Name	Shows the name by which the JBoss mail service is registered with the JNDI server.
Service - POP3 Server Host	Shows the host of the POP3 server for the JBoss mail service.
Service - SMTP Server Host	Shows the host of the SMTP server for the JBoss mail service.
Service - State	Shows the current state of the JBoss mail service. If the number is 3, then it is up.
Service - State (text)	Shows the state of the JBoss mail service.
Service - Store Protocol	Shows the protocol used by the JBoss mail service for receiving the messages.
Service - Transport Protocol	Shows the transport protocol used by the JBoss mail service for sending messages.

4.9 JBoss Message Driven Ejb Metrics

This metric category shows information for all the message driven EJBs loaded inside the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–9 JBoss Message Driven Ejb Metrics

Metric	Description
MDB - Instances Created Count	Shows the count of the instances created for this EJB by the EJB container.
MDB - Instances Removed Count	Shows the number of instance of this EJB that are removed from the EJB container.
MDB - Messages Count	Shows the number of messages received by this EJB.
MDB - Name	Shows name of the EJB.
MDB - Parent	Shows the parent or the jar file to which this EJB belongs.
Statistic Provider	Shows performance statistics support for this managed object (whether true or false).

4.10 JBoss Naming Service Metrics

This metric category provides information about the naming service of JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–10 JBoss Naming Service Metrics

Metric	Description
Listen Port	Shows the port on which the naming service of JBoss Application Server is running.
RMI Port	Shows the RMI port of the naming service of JBoss Application Server.
Service Name	Shows the name of the service monitored, that is, in this case it is NamingService only.
Service State	Shows the current state of the naming service of JBoss Application Server.

4.11 JBoss Protocols Ports Metrics

This category provides connection details for JBoss Application Server.

Table 4–11 JBoss Protocols Ports Metrics

Metric	Description
JBoss - Connection Linger Value	Represents linger value on the incoming connection for JBoss Application Server.
JBoss - Connection Timeout	Shows the timeout value on the incoming connection for JBoss Application Server.
JBoss - KeyStore File	Shows the KeyStore file for JBoss Application Server.
KeyStore Type	Shows the configured keystore type for JBoss Application Server.
JBoss - Protocol	Shows the coyote protocol handler in use for JBoss Application Server.
JBoss - Redirect Port	Shows the redirect port for non-SSL to SSL redirects for JBoss Application Server.
JBoss - Redirect Port Protocol	Shows the protocol name for this Connector (http, https) for JBoss Application Server.

4.12 JBoss Server Configuration Metrics

This category provides information about the configuration of JBoss Application Server.

Default Collection Interval — Every 30 minutes

Table 4–12 JBoss Server Configuration Metrics

Metric	Description
JBoss - Base Directory	Shows the base directory (directory 'server') for the JBoss Application Server.
JBoss - Configuration Name	Shows the name of the configuration that is used by JBoss Application Server. By default, JBoss Application Server comes with some configuration, for example, 'all', 'minimal', and so on.
JBoss - Deployment Filename	Shows the name of the main deployment file of JBoss Application Server. By default, it is jboss-service.xml. The configuration directory contains the "jboss-service.xml" bootstrap descriptor file for a given server configuration. This defines the core service-configurations that are fixed for the lifetime of the server.
JBoss - Home Directory	Shows the installation directory of the JBoss Application Server.
JBoss - Server Home Directory	Shows the configuration directory of JBoss Application Server that is running. If JBoss Application Server is running under 'minimal' configuration, then this metric shows the path to that configuration.

4.13 JBoss Server Details Metrics

This category provides server details for JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–13 JBoss Server Details Metrics

Metric	Description
Active Thread Count	This metric shows the active thread count for JBoss Application Server.
Active Thread Group Count	This metric shows the active thread group count for JBoss Application Server.
Available No. of Processors	Shows the available number of processors for JBoss Application Server.
Free Heap Size	Shows the free heap memory for JBoss Application Server.
Free Heap Size %	Shows the percentage of available heap memory out of total configured heap memory for JBoss Application Server.
Heap Size	Show the available heap memory (in MB) for JBoss Application Server.
Heap Size (MB)	Shows the available heap memory (in MB) for JBoss Application Server.
Host Name	Provides the host name on which JBoss Application Server is installed.
Java Vendor Name	Provides Java Vendor Name for JBoss Application Server.
Java Version	Shows the version of JDK/JRE for JBoss Application Server.
Java VM Name	Show the name of the Java Virtual Machine (JVM) used for JBoss Application Server.
Java VM Vendor Name	Show the Java VM vendor name with which this JBoss Application Server is running.
Java VM Version	Show the Java VM version with which JBoss Application Server is running.
Maximum Memory	Shows the maximum available memory on which JBoss Application Server is running.

Table 4–13 (Cont.) JBoss Server Details Metrics

Metric	Description
OS Architecture	Provides operating system architecture details on which JBoss Application Server is running.
OS Name	Provides operating system name on which JBoss Application Server is running.
OS Version	Show the operating system version on which JBoss Application Server is running.
Total Heap Size	Shows the total heap memory for JBoss Application Server.
Total Heap Size(MB)	Shows the total heap memory (in MB) for JBoss Application Server.

4.14 JBoss Server Metrics

This category provides information about JBoss Application Server.

Default Collection Interval — Every 30 minutes

Table 4–14 JBoss Server Metrics

Metric	Description
Host Address	Shows the IP address of the host on which the JBoss Application Serveris running.
Host Name	Shows the host name on which the JBoss Application Server is running.
Host Operating System	Shows the operating system of the host on which the JBoss Application Server is running.
JBoss - Build Date	Shows the build date for the JBoss Application Server.
JBoss - Build ID	Shows the build ID of the JBoss Application Server.
JBoss - Listen Port	Shows the naming service port or Java Naming Provider (JNP) port for the JBoss Application Server.
JBoss - Start Time	Shows date on which the JBoss Application Server was started.
JBoss - Version	Shows the version (build version) of the JBoss Application Server.
JBoss - Version Name	Shows the version name of the JBoss Application Server.

4.15 JBoss Servlets Raw Metrics

This metric category provides information for the servlets loaded inside the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–15 JBoss Servlets Raw Metrics

Metric	Description
Parent	Shows the parent or the application to which this servlet belongs.
Servlet - Active Requests / Sec	Shows rate of requests (per sec) on the servlet.
Servlet - Active Requests since last Upload	Shows the current active requests for the servlet.
Servlet - Active Requests, Response Time (Sec)	Shows the response time(per sec) for the servlet for handling those requests.

Table 4–15 (Cont.) JBoss Servlets Raw Metrics

Metric	Description
Servlet - Maximum Response Time	Shows the maximum processing time of a request (depicted by a long value).
Servlet - Minimum Response Time	Shows the minimum processing time of a request (depicted by a long value).
Servlet - Response Time Unit	Shows the unit of time for the requests.
Servlet - Total Requests	Shows the number of requests processed by this servlet.
Servlet - Total Response Time	Shows the average response time for the request.
Servlet Name	Shows the name of the JBoss servlet.
Statistic Description	Shows the description of the statistic instance.
Statistic Name	Shows the name of the statistic instance.
Statistic Provider	Shows the performance statistics support for this managed object (whether true or false).

4.16 JBoss Stateful Session EJB Metrics

This category shows information about the stateful EJBs loaded inside the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–16 JBoss Stateful Session EJB Metrics

Metric	Description
Stateful Ejb - Current Method Ready Count	Shows the count of the instances that are in the ready state in the EJB container.
Stateful Ejb - Current Passive Instances Count	Shows the number of instances (of this EJB) that are in the passive state.
Stateful Ejb - Instances Created Count	Shows the count of the instances created for this EJB by the EJB container.
Stateful Ejb - Instances Removed Count	Shows the number of instance of this EJB that are removed from the EJB container.
Stateful Ejb - Maximum Method Ready Count	Shows the maximum number of instances that can be in the ready state in the EJB container.
Stateful Ejb - Maximum Passive Instances Count	Shows the maximum number of instances that can be in the passive state.
Stateful Ejb - Minimum Method Ready Count	Shows the minimum number of instances that can be in the ready state in the EJB container.
Stateful Ejb - Minimum Passive Instances Count	Shows the minimum number of instances that can be in the ready state in the EJB container.
Stateful Ejb - Name	Shows the name of the EJB.
Stateful Ejb - Parent	Shows the parent or the JAR file to which this EJB belongs.
Statistic Provider	Shows performance statistics support for this managed object (whether true or false).

4.17 JBoss Stateless Session EJB Metrics

This category shows information about the stateless EJBs loaded inside the jboss server.

Default Collection Interval — Every 15 minutes

Table 4–17 JBoss Stateless Session EJB Metrics

Metric	Description
Stateless Ejb - Current Method Ready Count	Shows the count of the instances which are in the ready state in the EJB container.
Stateless Ejb - Instances Created Count	Shows the count of the instances created for this EJB by the EJB container.
Stateless Ejb - Instances Removed Count	Shows the count of the instances removed for this EJB by the EJB container.
Stateless Ejb - Maxinum Method Ready Count	Shows the maximum number of instances that can be in the ready state in the EJB container.
Stateless Ejb - Minimum Method Ready Count	Shows the minimum number of instances that can be in the ready state in the EJB container.
Stateless Ejb - Name	Shows the name of the EJB.
Stateless Ejb - Parent	Shows the parent or the JAR file to which this EJB belongs.
Statistic Provider	Shows performance statistics support for this managed object (whether true or false).

4.18 JBoss Top EJBs Metrics

This category shows information for the top performing EJBs of JBoss

Default Collection Interval — Every 15 minutes

Table 4–18 JBoss Top EJBs Metrics

Metric	Description
JBoss - Ejb Create Count	Shows the number of instances created for this EJB by the container.
JBoss - Ejb Remove Count	Shows the number of instances removed for this EJB by the container.
JBoss - Ejb Type	Shows the name of the EJB.

4.19 JBoss Top Servlets Metrics

This category gives information for the top performing servlets of the JBoss Application Server.

Table 4–19 JBoss Top Servlets Metrics

Metric	Description
JBoss - Servlet WAR Name	Shows the parent or application to which the servlet belongs.
JBoss - Web Component Type	Shows the name of the servlet.
Servlet - Active Requests / Sec	Shows rate of request/sec to this servlet.
Servlet - Active Requests since last Upload	Shows number of active requests for this servlet.

Table 4–19 (Cont.) JBoss Top Servlets Metrics

Metric	Description
Servlet - Active Requests, Response Time (Sec)	Shows the response time in sec for these active requests by the servlet.

4.20 JBoss Transactions Metrics

This category shows the information for the transaction manager service of the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–20 JBoss Transactions Metrics

Metric	Description
Committed last interval	Shows the number of transaction committed since last interval.
JBoss - Active Transactions Count	Shows the number of current active transactions.
JBoss - Transactions / Sec	Shows the rate of active transactions per second.
JBoss - Transactions Committed / Sec	Shows the rate of transactions committed per second.
JBoss - Transactions Committed Count	Shows the number of transaction that have been committed.
JBoss - Transactions Rolled Back / Sec	Shows the rate of transactions rolled back per second.
JBoss - Transactions RolledBack Count	Shows the number of transaction that have been rolled back.
JBoss - Transactions Timeout (Secs)	Shows the the time interval in secs after which the transactions will timeout.
Rolledback last interval	Shows the number of transactions rolled back since last interval.
Trans last interval	Shows the transaction that was made since last interval.

4.21 JBoss Web Applications Metrics

This category shows performance information about the Web application loaded inside the JBoss Application Server.

Default Collection Interval — Every 15 minutes

Table 4–21 JBoss Web Applications Metrics

Metric	Description
Web Apps - Active requests since last agent upload	Shows the current active requests for the application.
Web Apps - Average maximum response time	Shows the average maximum response time for the active requests.
Web Apps - Average minimum response time	Shows the average minimum response time for the active requests.
Web Apps - Average Response Time	Shows the average response given by these applications for the request.
Web Apps - Requests rate per second	Shows the current rate of requests (per sec) to this application.

Table 4–21 (Cont.) JBoss Web Applications Metrics

Metric	Description
Web Apps - Total Requests	Shows the total requests for the application.
Web Apps - Total Response Time	Shows the total time take for servicing those requests.
Web Apps - Total Response time for Active Requests	Shows the total response time for all the active requests.

4.22 JBoss WebService Service Metrics

This category provides information for the Web services running on the JBoss Application Server.

Table 4–22 JBoss WebService Service Metrics

Metric	Description
Service - Bind Address	Shows the IP address to which the Web service is bound.
Service - Host	Shows host name on which the Web service is running.
Service - Port	Shows the port on which the Web service is listening.
Service - State	Shows the state of the Web service as a number. Number "3" depicts that the service is up.
Service - State (text)	Shows the state of the Web service.

4.23 Resource Usage Metrics

This category provides metrics related to JBoss Application Server resource usage.

Table 4–23 Resource Usage

Metric	Description
Free Memory (%)	Shows the percentage of heap memory that is free and is not used by the Java Virtual Machine (JVM).
Maximum Memory Size (MB)	Provides total heap memory configured for JBoss Application Server.
Memory Usage (MB)	Shows the memory used by JBoss Application Server in MBs.

4.24 Response Metrics

This category provides response metrics for JBoss Application Server.

Table 4–24 Response Metrics

Metric	Description
UpDown Status	Shows whether the JBoss Application Server is up or down. If the value is 1, then JBoss is up. If the value is 0, then it is down. The upload frequency is after every sample. If the JBoss Application Server instance is down, then restart it from the scripts that come with JBoss Application Servers.

5

Microsoft .NET Framework Metrics

This chapter provides descriptions for all Microsoft .NET Framework metric categories, and tables list and describe associated metrics for each category. The tables also provide user actions if any of the metrics for a particular category support user actions.

5.1 .NET CLR Exceptions Metrics

The metrics in the .NET CLR Exceptions category provide information about the exceptions thrown by an application.

Default Collection Interval — Every 15 minutes

Table 5–1 .NET CLR Exceptions Metrics

Metric	Description and User Action
Exceptions Thrown	This counter indicates the total number of exceptions generated in managed code since the application restarted. Exceptions are very costly and can severely degrade your application performance. You should investigate your code for application logic that uses exceptions for normal processing behavior. `Response.Redirect`, `Server.Transfer`, and `Response.End` all cause a ThreadAbortException in ASP.NET applications. This counter value should be less than 5 percent of Request/sec for the ASP.NET application. If more than 1 request in 20 throw an exception, be attentive to these occurrences.
Exceptions Thrown Per Sec	This counter indicates the total number of exceptions generated per second in managed code. Exceptions are very costly and can severely degrade your application performance. You should investigate your code for application logic that uses exceptions for normal processing behavior. `Response.Redirect`, `Server.Transfer`, and `Response.End` all cause a ThreadAbortException in ASP.NET applications. This counter value should be less than 5 percent of Request/sec for the ASP.NET application. If more than 1 request in 20 throw an exception, be attentive to these occurrences.
Exceptions Thrown Since Last Upload	Displays the total number of exceptions thrown during the last metric collection. Exceptions are very costly and can severely degrade your application performance. You should investigate your code for application logic that uses exceptions for normal processing behavior. `Response.Redirect`, `Server.Transfer`, and `Response.End` all cause a Thread Abort Exception in ASP.NET applications. This counter value should be less than 5 percent of Request/sec for the ASP.NET application. If more than 1 request in 20 throw an exception, be attentive to these occurrences.

5.2 .NET CLR Interop Metrics

The metrics in the .NET CLR Interop category provide information about an application's interaction with COM components, COM+ services, and external type libraries.

Default Collection Interval — Every 15 minutes

Table 5–2 .NET CLR Interop Metrics

Metric	Description and User Action
CCWs	Displays the current number of COM callable wrappers (CCWs). A CCW is a proxy for a managed object being referenced from an unmanaged COM client. This counter indicates the number of managed objects referenced by unmanaged COM code. This value must be as low as possible. If this number is high, determine whether you can redesign this part of the application to reduce the number of transitions needed. If you see memory expanding (or not becoming freed when it should have been) and this counter is increasing, this may suggest that unmanaged code is holding on to some of your managed objects, thereby causing memory either not being freed or increasing.
Marshalling	Displays the total number of times arguments and return values have been marshalled from managed to unmanaged code, and vice versa, since the application started. This value must be as low as possible. If this number is high, determine whether you can redesign this part of the application to reduce the number of transitions needed.
Stubs	Displays the current number of stubs created by the common language runtime. Stubs are responsible for marshaling arguments and return values from managed to unmanaged code, and vice versa, during a COM interop call or a platform invoke call. This value must be as low as possible. For calls being made from managed and unmanaged code and vice-versa, this counter indicates the interaction between COM and managed code. If this number is high, determine whether you can redesign this part of the application to reduce the number of needed transitions.

5.3 .NET CLR JIT Metrics

The metrics in the .NET CLR Jit category provide information about Just-in-Time (JIT) information for the code that has just been compiled. JIT compilation is used to compile IL methods to native machine language immediately before execution of the methods.

Default Collection Interval — Every 15 minutes

Table 5–3 .NET CLR JIT Metrics

Metric	Description and User Action
IL Bytes Jitted	Displays the total number of Microsoft Intermediate Language (MSIL) bytes compiled by the Just-in-Time (JIT) compiler since the application started. A high value for this performance counter indicates performance overhead. The JIT is a runtime optimizing compiler. You can consider improving the startup time of client applications by compiling your application at install time, using the `NGEN.exe` utility.
Methods Jitted	Displays the total number of methods JIT-compiled since the application started. This counter does not include pre-JIT-compiled methods. A high value for this performance counter indicates performance overhead. The JIT is a runtime optimizing compiler. You can consider improving the startup time of client applications by compiling your application at install time, using the `NGEN.exe` utility.
Standard Jit Failures	Displays the peak number of methods the JIT compiler has failed to compile since the application started. A high value for this counter indicates a problem with the JIT compiler. This failure can occur if the IL cannot be verified, or if there was an internal error in the JIT compiler. Check the .NET framework setup to troubleshoot.

5.4 .NET CLR Loading Metrics

The metrics in the .NET CLR Loading category provide information assemblies, classes, and application domains that are loaded.

Default Collection Interval — Every 15 minutes

Table 5–4 .NET CLR Loading Metrics

Metric	Description and User Action
Load Failures	Displays the peak number of classes that have failed to load since the application started. Load failures can occur for many reasons, such as inadequate security or invalid format. Check your code to fix the problem.
Application Domains Loaded Since Last Upload	Displays the number of application domains loaded in this application during the last metric collection. Often, especially for security reasons, you cannot avoid using multiple AppDomains. However, doing so can limit performance at startup. You can reduce the impact of multiple App Domains by loading assemblies as domain neutral, enforcing efficient cross-AppDomain communication, using NeutralResourcesLanguageAttribute, and using serialization wisely.
Application Domains Unloaded Since Last Upload	Displays the number of application domains unloaded in this application during the last metric collection. Often, especially for security reasons, you cannot avoid using multiple AppDomains. However, doing so can limit performance at startup. You can reduce the impact of multiple App Domains by loading assemblies as domain neutral, enforcing efficient cross-AppDomain communication, using NeutralResourcesLanguageAttribute, and using serialization wisely.
Bytes in Loader Heap	Indicates the current size (in bytes) of committed memory by the class loader across all AppDomain(s). Committed memory is the physical memory for which space has been reserved in the paging file on disk. This counter must be in a steady state; otherwise, large fluctuations in this counter indicate there are too many assemblies loaded per AppDomain. Change the page file size accordingly to improve performance.
Current Application Domains	Displays the current number of application domains loaded in this application. The value should be same as the number of Web applications plus one. The additional application is the default application domain loaded by the ASP.NET worker process. Often, especially for security reasons, you cannot avoid using multiple AppDomains. However, doing so can limit performance at startup. You can reduce the impact of multiple App Domains by loading assemblies as domain neutral, enforcing efficient cross-AppDomain communication, using NeutralResourcesLanguageAttribute, and using serialization wisely.
Current Assemblies	Displays the current number of assemblies loaded across all application domains in the currently running application. It includes the Application Domains in the system. ASP.NET Web pages (.aspx files) and user controls (.ascx files) are "batch compiled" by default, which typically results in one to three assemblies, depending on the number of dependencies. An unusually high number of loaded assemblies can cause excessive memory consumption. Try to minimize the number of Web pages and user controls without compromising the efficiency of workflow. Assemblies cannot be unloaded from an application domain. To prevent excessive memory consumption, the application domain is unloaded when the number of recompilations (.aspx, .ascx, .asax) exceed the limit specified by `<compilation numRecompilesBeforeAppRestart=/>`. **Note:** If the `<%@ page debug=%>` attribute is set to true, or if `<compilation debug=/>` is set to true, batch compilation is disabled.
Current Classes Loaded	Displays the current number of classes loaded in all assemblies. If this value is too high, consider increasing the physical memory to improve performance.

Table 5–4 (Cont.) .NET CLR Loading Metrics

Metric	Description and User Action
Load Failures Since Last Upload	Displays the peak number of classes that have failed to load during the last metric collection. Load failures can occur for many reasons, such as inadequate security or invalid format. Check your code to fix the problem.
Time Loading Percent	Reserved for future use.
Total Application Domains	Displays the peak number of application domains loaded since the application started. Often, especially for security reasons, you cannot avoid using multiple AppDomains. However, doing so can limit performance at startup. You can reduce the impact of multiple AppDomains by loading assemblies as domain neutral, enforcing efficient cross-AppDomain communication, using NeutralResourcesLanguageAttribute, and using serialization wisely.
Total Application Domains Unloaded	Displays the peak number of application domains unloaded since the application started. Often, especially for security reasons, you cannot avoid using multiple AppDomains. However, doing so can limit performance at startup. You can reduce the impact of multiple App Domains by loading assemblies as domain neutral, enforcing efficient cross-AppDomain communication, using NeutralResourcesLanguageAttribute, and using serialization wisely.
Total Assemblies	Displays the total number of assemblies loaded since the application started. Try to minimize the number of Web pages and user controls without compromising the efficiency of workflow. Assemblies cannot be unloaded from an application domain. To prevent excessive memory consumption, the application domain is unloaded when the number of recompilations (.aspx, .ascx, .asax) exceed the limit specified by `<compilation numRecompilesBeforeAppRestart=/>`. **Note:** If the `<%@ page debug=%>` attribute is set to true, or if `<compilation debug=/>` is set to true, batch compilation is disabled.
Total Classes Loaded	Displays the cumulative number of classes loaded in all assemblies since the application started.

5.5 .NET CLR Locks and Threads Metrics

The metrics in the .NET CLR Locks and Threads category provide information about managed locks and threads that an application uses.

Default Collection Interval — Every 15 minutes

Table 5–5 .NET CLR Locks and Threads Metrics

Metric	Description and User Action
Contention Rate Per Sec	Displays the rate at which threads in the runtime unsuccessfully attempt to acquire a managed lock. Sustained nonzero values should cause concern. If this number is increasing, a bottleneck exists in the code. This area in the code is synchronized, so only one thread at a time enters it, but it is being "hammered" by multiple threads that are all attempting to get into this piece of code. To resolve this bottleneck, find this piece of code and determine how you can avoid this situation.
Contentions Since Last Upload	Displays the total number of times that threads in the runtime have attempted to acquire a managed lock unsuccessfully during the last metric collection. Sustained nonzero values should cause concern. If this number is increasing, a bottleneck exists in the code. This area in the code is synchronized, so only one thread at a time enters it, but it is being "hammered" by multiple threads that are all attempting to get into this piece of code. To resolve this bottleneck, find this piece of code and determine how you can avoid this situation.
Current Logical Threads	Displays the number of current managed thread objects in the application. This counter is not an average over time; it just displays the last observed value. This counter maintains the count of both running and stopped threads. A value that is too high may be a cause of concern.
Current Physical Threads	Displays the number of native operating system threads created and owned by the common language runtime to act as underlying threads for managed thread objects. A value that is too high indicates performance bottlenecks. If this value is too high, try to refactor the code to reduce the number of spawned threads to an optimal value.
Current Queue Length	Displays the total number of threads that are currently waiting to acquire a managed lock in the application. You may want to run dedicated tests for a particular piece of code to identify the average queue length for the particular code path. This helps you identify inefficient synchronization mechanisms.
Current Recognized Threads	Displays the number of threads that are currently recognized by the runtime; they have a corresponding .NET thread object associated with them. These threads are not created by the CLR. They are created outside the CLR, but have since run inside the CLR at least once. Only unique threads are tracked. Threads with same thread ID re-entering the CLR or recreated after thread exit are not counted twice.
Queue Length Peak	Displays the total number of threads that waited to acquire a managed lock since the application started. You may want to run dedicated tests for a particular piece of code to identify the average queue length for the particular code path. This helps you identify inefficient synchronization mechanisms.
Queue Length Per Sec	Displays the number of threads per second that are waiting to acquire a lock in the application. You may want to run dedicated tests for a particular piece of code to identify the average queue length for the particular code path. This helps you identify inefficient synchronization mechanisms.
Total Contentions	Displays the total number of times that threads in the runtime have attempted to acquire a managed lock successfully. Sustained nonzero values may be a cause of concern. You may want to run dedicated tests for a particular piece of code to identify the contention rate for the particular code path.
Total Recognized Threads	Displays the total number of threads that have been recognized by the runtime since the application started.

5.6 .NET CLR Memory Metrics

The metrics in the .NET CLR Memory category provide information about managed and unmanaged memory consumption.

Default Collection Interval — Every 15 minutes

Table 5–6 .NET CLR Memory Metrics

Metric	Description and User Action
Bytes in All Heaps	Displays the current memory allocated in bytes on the garbage collection heaps. This counter is the sum of four other counters — Gen 0 Heap Size, Gen 1 Heap Size, Gen 2 Heap Size, and Large Object Heap Size. The value of this counter is always less than the value of Process/Private Bytes, which also includes the native memory allocated for the process by the operating system. Private Bytes - # Bytes in all Heaps is the number of bytes allocated for unmanaged objects. If this counter continues to rise, there is a managed leak. Some managed objects are always being referenced and are never collected.
Pinned Objects	When .NET-based applications use unmanaged code, these objects are pinned in memory. That is, they cannot move around because the pointers to them would become invalid. These can be measured by this counter, which displays the number of pinned objects encountered in the last garbage collection. Too many pinned objects affect the performance of the garbage collector, because they restrict its ability to move objects and organize memory efficiently. You can also pin objects explicitly in managed code, such as reusable buffers used for I/O calls. Try to reduce the number of pinned objects in the code. Generally, this number shouldn't be too high if you don't call to unmanaged code too often. If this counter is increasing, it might suggest that you are pinning objects due to passing them into unmanaged code and not releasing the unmanaged code, or you explicitly pinned objects and forgot to unpin them. If this counter is increasing and the Virtual Bytes counter is also increasing, pin objects are being done too often and the GC cannot effectively compact the heap. This forces the heap to reserve additional virtual memory so the GC heap can expand and accommodate the requested needs of allocation.
Sink Blocks in Use	Displays the current number of synchronization blocks in use. Synchronization blocks are per-object data structures allocated for storing synchronization information. Synchronization blocks hold weak references to managed objects and must be scanned by the garbage collector. Synchronization blocks are not limited to storing synchronization information; they can also store COM interop metadata. This counter indicates performance problems with heavy use of synchronization primitives. If this counter continues to increase, examine all the locations where you are using synchronization objects and determine if they are really needed.
Time in Garbage Collection Percent	Displays the percentage of elapsed time that was spent performing a garbage collection since the last garbage collection cycle. This counter should average about 5 percent for most applications when the CPU is 70 percent busy, with occasional peaks. As the CPU load increases, so does the percentage of time spent performing garbage collection. Keep this in mind when you measure the CPU. The most common cause of a high value is making too many allocations, which may be the case if you are allocating on a per-request basis for ASP.NET applications. Study the allocation profile for your application if this counter shows a higher value.

5.7 .NET CLR Networking Metrics

The metrics in the .NET CLR Networking category provide information about data that an application sends and receives over the network.

Default Collection Interval — Every 15 minutes

Table 5–7 .NET CLR Networking Metrics

Metric	Description and User Action
Bytes Received	Displays the cumulative number of bytes received over all open socket connections. This number includes data and any protocol information that is not defined by TCP/IP. When tuning the network utilization of an application, this counter indicates the total traffic generated by all .NET-based applications. Note that these counters do not let you monitor a specific .NET-based application. However, they do not measure network traffic that is generated by applications that do not use the common language runtime.
Bytes Sent	Displays the cumulative number of bytes sent over all open socket connections since the process started. This number includes data and any protocol information that is not defined by TCP/IP. When tuning the network utilization of an application, this counter indicates the total traffic generated by all .NET-based applications. Note that these counters do not let you monitor a specific .NET-based application. However, they do not measure network traffic that is generated by applications that do not use the common language runtime.
Connections Established	Displays the cumulative number of socket connections established for this process since it started. A high value for this counter indicates a large number of users trying to use the application, which equates to high usage for the application.
Connection Established Since Last Upload	Displays the number of socket connections established for this process during the last metric collection. A high value for this counter indicates a large number of users trying to use the application, which equates to high usage for the application.
Datagrams Received	Displays the cumulative number of datagram packets received since the process started. When tuning the network utilization of an application, this counter indicates the datagram packets traffic received by all .NET-based applications.
Datagrams Sent	Displays the cumulative number of datagram packets sent since the process started. When tuning the network utilization of an application, this counter indicates the datagram packets traffic sent by all .NET-based applications.

5.8 .NET CLR Remoting Metrics

The metrics in the .NET CLR Remoting category provide information about .NET remoting performance and its various key performance counters. The throughput of the remote component can be measured using the Remote Calls Per Sec and Requests Per Sec counters.

Default Collection Interval — Every 15 minutes

Table 5–8 .NET CLR Remoting Metrics

Metric	Description and User Action
Channels	Displays the total number of remoting channels registered across all application domains since the application started. Channels transport messages to and from remote objects. Use TcpChannel for optimum performance. Use the TcpChannel in trusted server scenarios.
Context Bound Classes Loaded	Displays the current number of context-bound classes that are loaded. Classes that can be bound to a context are called context-bound classes. These classes are marked with context attributes, which provide usage rules for synchronization, thread affinity, transactions, and so forth.
Context Bound Objects Alloc Per Sec	Displays the number of context-bound objects allocated per second. Classes that can be bound to a context are called context-bound objects. These classes are marked with context attributes, which provide usage rules for synchronization, thread affinity, transactions, and so forth. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval.

Table 5–8 (Cont.) .NET CLR Remoting Metrics

Metric	Description and User Action
Context Proxies	Displays the total number of remoting proxy objects in this process since it started. A proxy object acts as a representative of the remote objects and ensures that all calls made on the proxy are forwarded to the correct remote object.
Contexts	Displays the current number of remoting contexts in the application. A context is a boundary containing a collection of objects with the same usage rules such as synchronization, thread affinity, transactions, and so forth.
Remote Calls Per Sec	Displays the number of remote procedure calls invoked per second. A remote procedure call is a call on any object outside the caller's application domain. This counter is not an average over time; it displays the difference between the values observed in the last two samples divided by the duration of the sample interval. More than one remote call may be required to complete a single operation. You need to divide the counter with the amount of requests to complete a single operation. This gives you the rate of operations completed per second. You might need to instrument your code to observe the request execution time.
Total Remote Calls	Displays the total number of remote procedure calls invoked since the application started. More than one remote call may be required to complete a single operation. You need to divide the counter with the amount of requests to complete a single operation. This gives you the rate of operations completed per second. You might need to instrument your code to observe the request execution time.

5.9 .NET CLR Security Metrics

The metrics in the .NET CLR Security category provide information about code access security checks and how they affect performance.

Default Collection Interval — Every 15 minutes

Table 5–9 .NET CLR Security Metrics

Metric	Description and User Action
Link Time Checks	Displays the total number of link-time code access security checks since the application started.
Percent Time In RT Checks	Displays the percentage of elapsed time spent performing runtime code access security checks since the last sample.
Percent Time Sig Authenticating	Reserved for future use.
Stack Walk Depth	Displays the depth of the stack during the last runtime code access security check. This counter is not an average. It just displays the last observed value.
Total Runtime Checks	Displays the total number of runtime code access security checks performed since the application started. When used together with the Stack Walk Depth counter, this counter indicates the performance penalty that your code incurs for security checks.

5.10 .NET Data Metrics

The metrics in the .NET Data category provide information about ADO.NET data access performance. These counters provide valuable information to effectively measure the data access performance by measuring the utilization and effectiveness of pooling and failed connects.

Default Collection Interval — Every 15 minutes

Table 5–10 .NET Data Metrics

Metric	Description and User Action
SQL Client Current Pooled and Non-Pooled Connections	Current number of connections, pooled or not. A very high value for this counter indicates unclosed connections in the application. Try closing all the connections when you are finished with them.
SQL Client Failed Commands	Total number of command executions that have failed for any reason. A high value for this counter is a cause of concern. Try examining the problem in the code to see if there are any setup issues.
SQL Client Failed Connects	Total number of connection open attempts that have failed for any reason. A high value for this counter is a cause of concern. Try examining the problem in the code to see if there are any setup issues.

5.11 NetCLR Response Metrics

The metric in the NetCLR Response category provides information about the status of the Microsoft .NET Framework.

Default Collection Interval — Every minute

Table 5–11 NetCLR Response Metrics

Metric	Description and User Action
Status	Displays the status of the Microsoft IIS 6.0 target. The status of the target is based on whether the service IISADMIN is running or stopped. If the status is shown as down, check whether the Microsoft IISADMIN service is running. If the status of the target is not shown properly, check whether the Agent is reachable or not.

6 Microsoft Active Directory Metrics

This chapter provides descriptions for all Microsoft Active Directory metric categories, and tables list and describe associated metrics for each category. The tables also provide user actions if any of the metrics for a particular category support user actions.

6.1 Address Book Metrics

The Address Book is a client for the Active Directory database. It performs lookups and search operations on the Active Directory database to look for details such as account email ID, and so forth. The metrics in the Address Book category provide information regarding these operations.

Default Collection Interval — Every 15 minutes

Table 6–1 Address Book Metrics

Metric	Description
Address Book Browse/Sec.	Shows the rate at which Address Book clients perform browse operations on the Active Directory.
Address Book Client Sessions	Shows the number of connected Address Book client sessions.
Address Book Lookups/Sec.	Shows the rate at which proxy clients perform search operations on the Active Directory.
Address Book Matches/Sec.	Shows the rate at which Address Book clients perform find operations on the Active Directory.
Address Book Property Reads/Sec.	Shows the rate at which Address Book clients perform property read operations on the Active Directory.
Address Book Reads/Sec.	Shows the rate at which Address Book clients perform read operations on the Active Directory.
Address Book Searches/Sec.	Shows the rate at which Address Book clients perform key search operations on the Active Directory.

6.2 All Errors from the Directory Service Event Log Metrics

This metric category shows all the errors currently listed in the windows event log file. It shows both system errors as well as application errors. Due to the bulk of data these metrics can show, these metrics are only shown as real-time metrics, and their data is not stored in the repository.

Default Collection Interval — Real-time only

Table 6–2 All Errors from the Directory Service Event Log Metrics

Metric	Description
Error Message	Description text of the error that is mentioned in the event log file.
Event Type	Indicates whether this is a warning or an error.
Record Number	Every error generated has an event ID or record number. You can search more information on the web and the MSN technet website using this event ID.
Source Name	Component that generated the error or warning, such as NTDS Inter-site messaging.
Time Written	Date and time when the error was generated.

6.3 Database Log Files Metrics

All the events (whether search, lookup, update, and so forth) performed on the Active Directory database are stored in log files. Four standard log files are created when Active Directory is installed. Over a period of time, these log files can become larger and need to be cleaned for maintenance. The metrics in this category provide information regarding these log files, consisting of their current size and the free space remaining in the drive where they are installed.

Default Collection Interval — Every 2 hours

Table 6–3 Database Log Files Metrics

Metric	Description and User Action
Database Log Drive Free Space	Free space remaining on the drive where the log files are stored.
Database Log Drive Free Space (GB)	Free space (in GB) remaining on the drive where the log files are stored.
Database Log File Size (GB)	Size of the log file in gigabytes. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements for the adequate file size.
Database Log File Size (MB)	Size of the log file in megabytes.
Database Log File Size Bytes	Size of the log file in bytes.
Database Log Files Location	Path of the log file on the operating system drive.

6.4 Directory Database Metrics

The metrics in this category show the size of the Active Directory database in bytes and MBs, and also shows the free space remaining on the operating system drive where the Active Directory database file is stored.

Default Collection Interval — Every 2 hours

Table 6–4 Directory Database Metrics

Metric	Description and User Action
Database Drive Free Space	Free space remaining on the drive where the Active Directory database file is stored.
Database Drive Free Space (GB)	Free space remaining (in GB) on the drive where the Active Directory database file is stored.
Database File Name	Name of the Active Directory database file.

Table 6–4 (Cont.) Directory Database Metrics

Metric	Description and User Action
Database File Size (Bytes)	Database file size in bytes.
Database File Size (MB)	Database file size in megabytes.
Database File Size (GB)	Database file size in gigabytes. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements for the adequate file size.

6.5 Directory Replication Monitoring Metrics

The metrics in this category show the status of the Active Directory replication monitoring by depicting the bytes replicated, errors that have occurred, and so forth. The replication subsystem maintains data consistency across all domain controllers in a domain of Active Directory.

Default Collection Interval — Every 15 minutes

Table 6–5 Directory Replication Monitoring Metrics

Metric	Description and User Action
Highest USN Committed (High Part)	Shows the high-order 32 bits of the highest update sequence number (USN) committed on the directory system agent (DSA).
Highest USN Committed (Low Part)	Shows the low-order 32 bits of the highest USN committed on the DSA.
Highest USN Issued (High Part)	Shows the high-order 32 bits of the highest USN issued on the DSA.
Highest USN Issued (Low Part)	Shows the low-order 32 bits of the highest USN issued on the DSA.
Inbound Full Sync Objects Remaining	Shows the number of objects remaining until the full synchronization is completed (while replication is done).
Inbound Object Updates Remaining in Packet	Shows the number of object updates received in the current directory replication update packet that have not yet been applied to the local server.
Inbound Objects Applied/Sec	Shows the rate at which replication updates received from replication partners are applied by the local directory service. This counter excludes changes that are received but not applied (because, for example, the change has already been made). This indicates how much replication update activity is occurring on the server because of changes generated on other servers. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements. If the current value is less than the warning and threshold value you define, the replication for objects applied per second is slow.
Inbound Objects Filtered/Sec	Shows the number of objects received from inbound replication partners that contained no updates that needed to be applied.
Inbound Objects/Sec	Shows the number of objects received from neighbors through inbound replication. A neighbor is a domain controller from which the local domain controller replicates locally.
Inbound Properties Applied/Sec	Shows the number of properties updated because of the incoming property winning the reconciliation logic that determines the final value to be replicated.
Inbound Properties Filtered/Sec	Shows the number of property changes received during the replication that has already been seen.

Table 6–5 (Cont.) Directory Replication Monitoring Metrics

Metric	Description and User Action
Inbound Properties Total/Sec	Shows the total number of object properties received from inbound replication partners.
Inbound Values (DNs Only)/Sec	Shows the number of object property values received from inbound replication partners that are distinguished names (DNs) that reference other objects. DN values, such as group or distribution list memberships, are generally more expensive to apply than other types of values.
Inbound Values Total/Sec	Shows the total number of object property values received from inbound replication partners. Each inbound object has one or more properties, and each property has zero or more values. Zero values indicate property removal.
Inbound Bytes Compressed (Between Sites After Compression)/Sec	Shows the compressed size of inbound compressed replication data (size after compression from DSAs in other sites.)
Inbound Bytes Compressed (Between Sites Before Compression)/Sec	Shows the original size of inbound compressed replication data (size before compression from DSAs in other sites).
Inbound Bytes Not Compressed (Within Site)/Sec	Shows the number of incoming bytes replicated that were not compressed at the source (that is, from DSAs in the same site).
Inbound Bytes Total/Sec	Shows the total number of bytes replicated in. This counter is the sum of the number of uncompressed bytes (never compressed) and the number of compressed bytes (after compression). The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements. If the current value is less than the warning and critical threshold value you defined, the replication traffic over time is slow.
Outbound Bytes Compressed (Between Sites After Compression)/Sec	Shows the compressed size of outbound compressed replication data (size after compression from DSAs in other sites.)
Outbound Bytes Compressed (Between Sites Before Compression)/Sec	Shows the original size of outbound compressed replication data (size before compression from DSAs in other sites.)
Outbound Bytes Not Compressed (Within Site)/Sec	Shows the number of bytes replicated out that were not compressed (that is, from DSAs in the same site).
Outbound Bytes Total/Sec	Shows the total number of bytes replicated out that were not compressed (that is, from DSAs in the same site). The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements. If the current value is less than the warning and critical threshold value you defined, the replication traffic over time is slow.
Outbound Objects Filtered/Sec	Shows the number of objects that were determined by outbound replication to have no updates that the outbound partner did not already have.
Outbound Objects/Sec	Shows the number of objects replicated out. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements. If the current value is less than the warning and critical threshold value you defined, the replication traffic over time is slow.

Table 6–5 (Cont.) Directory Replication Monitoring Metrics

Metric	Description and User Action
Outbound Properties/Sec	Shows the number of properties replicated out.
Outbound Values (DNs Only)/Sec	Shows the number of object property values containing DNs sent to outbound replication partners. DN values, such as group or distribution list memberships, are generally more expensive to read than other kinds of values.
Outbound Values Total/Sec	Shows the number of object property values sent to outbound replication partners.
Pending Replication Synchronizations	Shows the number of directory synchronizations queued for this server but not yet processed.
Pending Replication Operations	Shows the number of directory operations not yet processed.
Synchronization Failures on Schema Mismatch	Shows the number of synchronization requests made to neighbors that failed because their schema is not synchronized.
Synchronization Requests Made	Shows the number of synchronization requests made to neighbors.
Synchronization Requests Successful	Shows the number of synchronization requests made to neighbors that were successfully returned.

6.6 Directory Service Metrics

The metrics in this category provide information about the Active Directory services.

Default Collection Interval — Every 15 minutes

Table 6–6 Directory Service Metrics

Metric	Description and User Action
Client Binds in the Last Interval	Client bind operations performed since the last agent upload.
Directory Client Binds/Sec	Shows the number of `ntdsapi.dll` binds per second serviced by this domain controller. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Directory Client Translations/Sec	Shows the number of `ntdsapi.dll` name translations per second serviced by this domain controller.
Directory Name Cache Hits/Sec	Shows the percentage of directory object name components lookups that are satisfied from the DSA's name cache.
Directory Percentage Reads by DRA	Shows the percentage of reads on the directory by replication.
Directory Percentage Reads by KCC	Shows the percentage of reads performed by the knowledge consistency checker on the directory.
Directory Percentage Reads by LSA	Shows the percentage of reads performed by the Local Security Authority (LSA) on the directory.
Directory Percentage Reads by NSPI	Shows the percentage of reads performed by the Name Service Provider Interface (NSPI) on the directory.
Directory Percentage Reads by NTDS API	Shows the percentage of reads performed by the name service directory APIs on the directory.
Directory Percentage Reads by Others	Shows the percentage of reads performed by other components on the directory.

Table 6–6 (Cont.) Directory Service Metrics

Metric	Description and User Action
Directory Percentage Reads by SAM	Shows the percentage of reads performed by the Security Authentication Server (SAM) on the directory.
Directory Percentage Searches by DRA	Shows the percentage of searches performed by the replication service on the directory.
Directory Percentage Searches by KCC	Shows the percentage of searches performed by the Knowledge Consistency Checker (KCC) on the directory.
Directory Percentage Searches by LDAP	Shows the percentage of searches performed by Lightweight Directory Access Protocol (LDAP) on the directory.
Directory Percentage Searches by LSA	Shows the percentage of searches performed by the Local Security Authority (LSA) on the directory.
Directory Percentage Searches by NSPI	Shows the percentage of searches performed by the Name Service Provider Interface (NSPI) on the directory.
Directory Percentage Searches by NTDS API	Shows the percentage of searches performed by the NTDS API on the directory.
Directory Percentage Searches by Others	Shows the percentage of searches performed by other components on the directory.
Directory Percentage Searches by SAM	Shows the percentage of searches performed by the Security Authentication Server (SAM) on the directory.
Directory Percentage Writes by DRA	Shows the percentage of write operations performed by the replication service on the directory.
Directory Percentage Writes by KCC	Shows the percentage of write operations performed by the Knowledge Consistency Checker (KCC) on the directory.
Directory Percentage Writes by LDAP	Shows the percentage of write operations performed by Lightweight Directory Access Protocol (LDAP) on the directory.
Directory Percentage Writes by LSA	Shows the percentage of write operations performed by the Local Security Authority (LSA) on the directory.
Directory Percentage Writes by NSPI	Shows the percentage of write operations performed by the Name Service Provider Interface (NSPI) on the directory.
Directory Percentage Writes by NTDS API	Shows the percentage of write operations performed by the NTDS API on the directory.
Directory Percentage Writes by Others	Shows the percentage of write operations performed by other components on the directory.
Directory Percentage Writes by SAM	Shows the percentage of write operations performed by the Security Authentication Server (SAM) on the directory.
Directory Reads in the Last Interval	Number of read operations since the last agent upload.
Directory Reads/Sec	Rate of directory reads per second.
Directory Search Sub Operations/Sec	Rate of directory search sub operations per second.
Directory Searches in the Last Interval	Number of search operations since the last agent upload.
Directory Searches/Sec	Number of directory searches per second. The default warning and critical threshold values for this metric are set to an ‘UnDefined’ value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Directory Security Description Sub Operations/Sec	Shows the number of security descriptor propagation suboperations per second. One security descriptor propagation operation is comprised of many suboperations.

Table 6–6 (Cont.) Directory Service Metrics

Metric	Description and User Action
Directory Server Binds/Sec	Shows the number of domain controller-to-domain controller binds per second that are serviced by this domain controller. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Directory Server Name Translations/Sec	Shows the number of domain controller-to-domain controller name translations per second that are serviced by this domain controller.
Directory Writes in the Last Interval	Number of write operations on the directory since the last agent upload.
Directory Writes/Sec	Rate of directory write operations per second.
Monitor List Size	Shows the number of requests to be notified when objects are updated that are currently registered with the DSA.
Notify Queue Size	Shows the number of pending update notifications that have been queued but not yet transmitted to clients.
Percent Reads from NTDS API	Percent of read operations on the directory by the directory service APIs.
Search Sub Operations in the Last Interval	Search Sub Operations in the Last Interval.
Server Binds in the Last Interval	Server Binds in the Last Interval Server bind operations performed since the last agent uploaded.
Threads in Use	Shows the current number of threads in use by the directory service. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

6.7 Event Log File Information Metrics

Event Log files contain the errors, warnings, and information about the messages generated for the various components. The metrics in this category provide information about the general windows event log files, such as SYSTEM.

Default Collection Interval — Every 48 hours

Table 6–7 Event Log File Information Metrics

Metric	Description
File Size (MB)	Event log file size in megabytes.
File Size Bytes	Byte Event log file size in bytes.
Last Accessed	Time when the file was last accessed.
Last Modified	Time when the file was last modified.
Max FileSize Allowed (MB)	Maximum file size allowed by the OS for the event log file in megabytes.
Max FileSize Allowed Bytes	Maximum file size allowed by the OS for the event log file in bytes.
Status	Current status of the event log file.

6.8 Flexible Single Master Operations (FSMO) Metrics

The metrics in this category provide the names of the various domain controllers in the forest and their role.

Default Collection Interval — Every 24 hours

Table 6–8 Flexible Single Master Operations Metrics

Metric	Description
Domain Controller	Domain controller performing that role. (This is the DSN name of the machine.)
Role	Role played by the domain controller.

6.9 Important Services Metrics

The metrics in this category provide information about all the important services that Active Directory depends upon.

Default Collection Interval — Every 11 minutes

Table 6–9 Important Services Metrics

Metric	Description
Display Name	Display name of the service.
Path	Path of the executable to invoke the service.
State	Current state of the service, such as Running.
Status	Indicates whether the status of the service is "OK" or not.

6.10 Knowledge Consistency Checker (KCC) Metrics

The metrics in this category provide information about the Knowledge Consistency Checker (KCC), which is part of the replication subsystem.

Default Collection Interval — Every 15 minutes

Table 6–10 Knowledge Consistency Checker Metrics

Metric	Description and User Action
KDCA Requests	Shows the number of Authentication Server (AS) requests serviced by the Kerberos Key Distribution Center (KDC) per second. The client uses AS requests to obtain a ticket-granting ticket.
KDCA Requests in the Last Interval	Shows the number of KDCA requests since the last Agent upload.
KDCA Requests/Sec	Rate of KDCA requests per second. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
KDCTGS Requests	Rate of KDCTGS requests per second. Shows the number of Ticket Granting Server (TGS) requests serviced by the KD C per second. The client uses TGS requests to obtain a ticket to a resource.
KDCTGS Requests/Sec	Rate of KDCTGS requests per second.
Kerberos Authentication in the Last Interval	Kerberos authentications since the last agent upload.
Kerberos Authentications	Shows the number of times per second that clients use a ticket to this domain controller to authenticate to this domain controller.
Kerberos Authentications/Sec	Rate of Kerberos authentications per second. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

6.11 Latest Errors and Warnings Metrics

The metrics in the this category provide information regarding the latest errors and warnings generated in the windows event log since the last agent upload time.

Default Collection Interval — Every 2 hours

Table 6–11 Latest Errors and Warnings Metrics

Metric	Description
Category	Error under a category type, such as a system error.
User	User under whose ID the error was generated.
Windows Event Severity	Severity of the error or warning.

6.12 Lightweight Directory Access Protocol (LDAP) Metrics

The metrics in the this category provide performance information for the directory LDAP server.

Default Collection Interval — Every 15 minutes

Table 6–12 Lightweight Directory Access Metrics

Metric	Description and User Action
Active Threads	Shows the current number of threads in use by the LDAP subsystem of the local directory service. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Client Connections in the Last Interval	LDAP connections made since the last upload to the repository.
Client Sessions	Shows the number of currently connected LDAP client sessions. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
LDAP UDP Operations/Sec	Shows the number of User Datagram Protocol (UDP) operations that the LDAP server is processing per second.
LDAP Client Connections/Sec	Shows the number of client connections made using the LDAP protocol per second.
LDAP Directory Searches/Sec	Shows the rate at which LDAP clients perform search operations. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
LDAP Directory Writes/Sec	Shows the rate at which LDAP clients perform write operations.
LDAP New Connections/Sec	Shows the rate at which new client connections are made per second. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
LDAP Server Binds/Sec	Shows the rate of server bind operations per second. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

Table 6–12 (Cont.) Lightweight Directory Access Metrics

Metric	Description and User Action
LDAP Writes in the Last Interval	Shows the write operations performed on the directory using LDAP since the last Agent upload.
New Connections in the Last Interval	Shows the new connections made on the directory using LDAP since the last Agent upload.
Searches in the Last Interval	Shows the search operations performed on the directory using LDAP since the last Agent upload.
Server Bind Time (Sec)	Shows the number of milliseconds taken for the last successful LDAP bind. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Server Binds in the Last Interval	Shows the server bind operations since the last Agent upload.
UDP Operations in the Last Interval	Shows the UDP operations since the last Agent upload.

6.13 Local Security Authentication Server (LSASS) Metrics

The Local Security Authority (LSA) is a protected subsystem that maintains security for the local computer. The metrics in this category provide information about this process, such as the bytes used by the LSA.

Default Collection Interval — Every 15 minutes

Table 6–13 Local Security Authentication Server Metrics

Metric	Description and User Action
Handles Count	Number of current handles.
Input/Output Data Bytes per Second	Number of bytes input/output by this process. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Name	Name of the LSASS process.
Percent Processor Time Used	Shows the percentage of CPU time used by this process.
Private Bytes Used	CPU memory used by the process in bytes.
Private Bytes Used (KB)	CPU memory used by the process in kilobytes.
Private Bytes Used (MB)	CPU memory used by the process in megabytes.
Threads Count	Number of current threads. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

6.14 Local Security Authentication Server CPU Overload Metrics

LSASS is the most memory-intensive process of Active Directory. If you calculate the CPU usage of this process, it is almost equivalent to calculating the overall load of Active Directory on the machine on which it is installed. The metrics in this category provide the percentage of load applied on the machine by the Active Directory.

Default Collection Interval — Every 10 minutes

Table 6–14 Local Security Authentication Server CPU Overload Metrics

Metric	Description and User Action
CPU Load Percentage	Percentage of CPU memory consumed by the LSASS process. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

6.15 Lost and Found Objects Metrics

The metrics in this category provide information the number of objects in the Lost and Found container of Active Directory.

Default Collection Interval — Every 2 hours

Table 6–15 Lost and Found Objects Metrics

Metric	Description and User Action
Lost and Found Objects Count	Number of objects in the Lost and Found container. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

6.16 NT File Replication Service (NtFrs) Metrics

The metrics in the NT File Replication Service category provide information about the NT file replication service process.

Default Collection Interval — Every 15 minutes

Table 6–16 NT File Replication Service Metrics

Metric	Description
Handles Count	Number of current handles.
Input/Output Data Bytes per Second	Number of bytes input/output by this process.
Name	Name of the NtFrs process.
NtFrs CPU Usage %	Percentage of CPU used by the NtFrs process.
NtFrs Memory Usage (KB)	CPU memory used by the process in kilobytes.
NtFrs Memory Usage (MB)	CPU memory used by the process in megabytes.
Percent Processor Time Used	Shows the percentage of CPU time used by this process.
Private Bytes Used	CPU memory used by the process in bytes.
Threads Count	Number of current threads for this process.

6.17 Response Metrics

The metrics in this category provide the overall status or health of the Active Directory.

Default Collection Interval — Every minute

Table 6–17 Response Metrics

Metric	Description and User Action
Status	Indicates whether the Active Directory is in a healthy state or is down. The default warning and critical threshold values for this metric are set to 1. If the value is 0, the target is down, and if the value is 1, the target is up and running. If the value is anything other than 0 or 1, the target is either unavailable or unreachable.

6.18 Security Accounts Manager (SAM) Metrics

The metrics in the this category provide information about the performance of the Security Accounts Manager.

Default Collection Interval — Every 15 minutes

Table 6–18 Security Accounts Manager Metrics

Metric	Description and User Action
Account Group Evaluation Latency	The time taken by SAM to evaluate an account group.
Computer Creations in the Last Interval	The computer create attempts since the last Agent upload.
Computer Creations/Sec Inc Req	Number of SAM create machine attempts per second.
Display Information Queries/Sec	Number of SAM query displays per second.
Enumerations in the Last Intervals	SAM enumerations since the last Agent upload.
Global Catalog Evaluations in the Last Interval	SAM global catalog evaluations in the last interval.
Global Catalog Evaluations/Sec	Number of SAM global catalog evaluations per second. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Global Group Member Evaluations/Sec	Number of SAM account group membership evaluations per second.
Machine Create Attempts/Sec	Number of SAM create machine attempts per second.
Member Changes in the Last Interval	SAM member changes since the last Agent upload.
Member Changes/Sec	Number of SAM membership changes.
Non Transactional Member Evaluations/Sec	Number of SAM nontransitive membership evaluations per second.
Password Changes in the Last Interval	SAM password changes since the last Agent upload.
Password Changes/Sec	Number of SAM password changes per second. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
SAM Enumerations/Sec	Number of SAM enumerations per second.
Successful User Creations/Sec	Number of users successfully created per second.

Table 6–18 (Cont.) Security Accounts Manager Metrics

Metric	Description and User Action
Transactional Member Evaluations/Sec	Number of SAM transitive membership evaluations per second.
Universal Group Members/Sec	Number of SAM universal group membership evaluations per second.
User Create Attempts in the 1st intv	SAM user create attempts since the last Agent upload.
User Create Attempts/Sec	Number SAM create user attempts per second.

6.19 Trust Information Metrics

Under Active Directory, various levels of trust can be defined between different domains in a forest. This trust can be unidirectional or bidirectional. The trust defines the boundary for the resources that can be accessed, or operations that can be performed between different domains in a forest. For instance, if a user from `domain1.company.com` wants to access some documents in `domain2.company.com`, a trust can be defined for this.

Default Collection Interval — Every 17 minutes

Table 6–19 Trust Information Metrics

Metric	Description and User Action
Trust Direction	Indicates whether the trust is unidirectional or bidirectional.
Trust is OK	Indicates whether the current status of the trust is OK or not.
Trust Status	Indicates the current status, such as running, stopped, and so forth. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

7

Microsoft BizTalk Server Metrics

The Microsoft BizTalk Server 2004 management plug-in for Oracle Enterprise Manager provides performance monitoring of BizTalk Server using performance counters and application events.

This chapter provides descriptions for all Microsoft BizTalk Server metric categories, and tables list and describe associated metrics for each category. The tables also provide user actions if any of the metrics for a particular category support user actions.

7.1 BizTalk BAS Inbox Document Library Metrics

A document library in SharePoint Services that BAS uses for receiving messages and documents from a running business process (BizTalk Orchestration) is called an Inbox. This is conceptually analogous to the Inbox used in a typical e-mail system.

The metrics in this category provide performance-related information about the Inbox.

Table 7–1 BizTalk BAS Inbox Document Library Metrics

Metric	Description
Total Fallbacks to Orphaned Messages	Total number of messages redirected from the partner index library, Inbox folder, to the Orphaned Messages document library because of an issue with the partner.
Total Inbox Failures	Total number of messages submitted by the Microsoft BizTalk Server that failed to persist into Windows SharePoint Services. The BizTalk Server resubmits these messages until they persist successfully or until the number of retries is exceeded.
Total Non Office Documents Created	Total number of messages successfully saved as is in the partner index library, Inbox folder, because of failure(s) to transform into InfoPath document(s).
Total Office Documents Created	Total number of InfoPath documents created in the partner document library, Inbox folder.

7.2 BizTalk BAS Outbox Document Library Metrics

A document library in SharePoint Services that BAS uses for sending messages and documents to a running business process (BizTalk Orchestration) is called an Outbox. This is conceptually analogous to the Outbox used in a typical e-mail system.

The metrics in this category provide information about the Outbox.

Table 7–2 BizTalk BAS Outbox Document Library Metrics

Metric	Description
Total Documents Not Sent	Total number of documents that the BizTalk SharePoint Messaging Adapter service failed to process and send to the BizTalk Server.
Total Documents Sent	Total number of documents that the BizTalk SharePoint Messaging Adapter service has successfully processed and asynchronously sent to the BizTalk Server.

7.3 BizTalk BAS Sent Items Document Library Metrics

A document library in SharePoint Services that BAS uses for archiving messages and sent documents is referred to as Sent Items. The Sent Items folder serves as the storing mechanism for tracking the sent business documents for future reference and auditing needs.

The metrics in this category provide information about Sent Items.

Table 7–3 BizTalk BAS Sent Items Document Library Metrics

Metric	Description
Total Documents Moved	Total number of documents successfully moved from the Outbox document library to the Sent Items folder of the partner document library.
Total Documents Moved with Rename	Total number of documents successfully moved from the Outbox document library to the partner document library, Sent Items folder, after they have been renamed because of a name collision.
Total Move Failures	Total number of messages submitted by the BizTalk Server that failed to be moved from the Outbox.
Total Move Fallbacks to Orphaned Messages	Total number of move operations that were redirected from the partner document library, Sent Items folder, to the Orphaned Messages document library because of an issue with the partner.
Total Non Office Documents Recreated	Total number of documents successfully recreated as is in the Sent Items folder because of failure(s) to transform them into InfoPath documents.
Total Office Documents Recreated	Total number of InfoPath documents recreated in the Sent Items folder of partner document libraries.
Total Recreated Fallbacks to Orphaned Messages	Total number of messages redirected to the Orphaned Messages document library because of an issue with the partner. The BizTalk Server resubmits these messages until they persist successfully or until the number of retries is exceeded.
Total Sent Items Failures	Total number of confirmation messages submitted by the BizTalk Server that failed to persist into Windows SharePoint Services.

7.4 BizTalk BAS TPM Management Web Service Metrics

Trading Partner Management (TPM) tools are used to create and update partner profiles to:

- Manage key information about addresses, contacts, and so forth
- Create groups of partners based on business needs
- Create and manage documents with partners
- Analyze and track the business process with partners.

The metric in this category provides information about the Trading Partner Management Web Service.

Table 7–4 BizTalk BAS TPM Management Web Service Metrics

Metric	Description
Get Parameter Calls	Number of GetParameter calls.

7.5 BizTalk BAS TPM Publishing Web Service Metrics

Trading Partner Management (TPM) tools are used to create and update partner profiles to:

- Manage key information about addresses, contacts, and so forth.
- Create groups of partners based on business needs
- Create and manage documents with partners
- Analyze and track the business process with partners.

The metrics in this category provide information about publishing Web Service.

Table 7–5 BizTalk BAS TPM Publishing Web Service Metrics

Metric	Description
Get Parameters Value Calls	Number of GetParameterValue calls.

7.6 BizTalk Enterprise SSO Metrics

BizTalk's Enterprise Single Sign-On (SSO) is a standalone service that enables you to map a Windows user account to one or more alternative Windows or non-Windows accounts. These accounts are mapped per application to securely access applications that require credentials other then those originally provided by the end user.

The metrics in this category provide information about Single Sign On.

Table 7–6 BizTalk Enterprise SSO Metrics

Metric	Description
Credential Cache Size	Total number of credentials in cache. The credential cache size provides data about user login activity.
Get Configuration Info	
Get Configuration Info Per Sec	Configuration infomation accessed per second.
Get Credentials	
Get Credentials Per Sec	Credentials accessed per second.
Issue Ticket	
Issue Ticket Per Sec	Tickets issued per second.
Redeem Ticket	
Redeem Ticket Per Sec	Tickets redeemed per second.
Validate and Redeem Ticket	
Validate and Redeem Ticket Per Sec	Tickets validated and redeemed per second.

7.7 BizTalk Error Events Metrics

The metric in this category provides information about ERROR events the application raises.

Table 7–7 BizTalk Error Events Metrics

Metric	Description
Windows Event Severity	Severity of the ERROR event.

7.8 BizTalk Human Workflow Service Metrics

Human Workflow Services (HWS) is a component of BizTalk Server 2004 that enables the creation of both a priori and ad hoc workflow models. This provides support for both well known constraint-driven models of processing, as well as the ability to compose steps into a workflow on the fly.

The metrics in this category provide information about HWS.

Table 7–8 BizTalk Human Workflow Service Metrics

Metric	Description
Activity Flows Retrieved	Number of requests for retrieving activity flow information.
New Action Added	Number of requests for instantiating actions or activity blocks.
New Activity Flows	Number of requests for creating new activity flows.
Steps Retrieved	Number of requests for retrieving step information.
Tasks Retrieved	Number of requests for retrieving task information.

7.9 BizTalk Messaging Documents Metrics

The metrics in this category provide information about Messaging Engines that process documents.

Table 7–9 BizTalk Messaging Documents Metrics

Metric	Description
Documents Processed	Processed BizTalk Messaging Documents.
Documents Processed Per Sec	Processed BizTalk Messaging Documents per second.
Documents Received	Received BizTalk Messaging Documents.
Documents Received Per Sec	Received BizTalk Messaging Documents per second.
Documents Suspended	Suspended BizTalk Messaging Documents.
Documents Suspended Per Sec	Suspended BizTalk Messaging Documents per second.
Process ID	Process ID for BizTalk Messaging Documents.

7.10 BizTalk Response Metrics

The metrics in this category provide information about BizTalk status.

Table 7–10 BizTalk Response Metrics

Metric	Description
Status	0 or 1 based on the status of the BTSSvc service as Running or Stopped.

7.11 BizTalk Tracking Data Decode Service Metrics

The metrics in this category provide information about the Tracking Data Decode Service (TDDS), also known as the BAM Event Bus Service.

Table 7–11 BizTalk Tracking Data Decode Service Metrics

Metric	Description
Batches Being Processed	Currently processing BizTalk TDDS batches.
Batches Committed	Committed BizTalk TDDS batches.
Events Being Processed	Currently processing BizTalk TDDS events.
Events Committed	Committed BizTalk TDDS events.
Records Being Processed	Currently processing BizTalk TDDS records.
Records Committed	Committed BizTalk TDDS records.
Total Batches	BizTalk TDDS total batches.
Total Events	BizTalk TDDS total events.
Total Failed Batches	BizTalk TDDS total failed batches.
Total Failed Events	BizTalk TDDS total failed events.
Total Records	BizTalk TDDS total records.

7.12 BizTalk Warning Events Metrics

The metrics in this category provide information about WARNING events the application raises.

Table 7–12 BizTalk Warning Events Metrics

Metric	Description
Description	Description of the WARNING event.
Windows Event Severity	Severity of the event.

7.13 Orchestrations Metrics

BizTalk Orchestration services provide a development and execution environment that integrates loosely-coupled, long-running business processes, both within and between businesses. It allows business analysts and developers to visually model their business processes using a Visio-like design environment and then bind that visual representation to its physical implementation.

The metrics in this category provide information about the Orchestration Service.

Table 7–13 Orchestrations Metrics

Metric	Description
Allocated Private Memory (MB)	Megabytes of allocated private memory for the host instance.
Allocated Virtual Memory (MB)	Megabytes reserved for virtual memory for the host instance.
Dehydratable Orchestrations	Number of orchestrations instances that can be dehydrated that are currently hosted by the host instance.
Dehydrating Orchestrations	Number of orchestrations that are in the process of dehydrating.
Dehydration Cycle in Progress	Indicates whether there is a dehydration cycle currently in progress.
Dehydration Cycles	Number of completed dehydration cycles.
Dehydration Threshold	Number in milliseconds that determines how aggressively orchestrations are being dehydrated. If the orchestration engine predicts that an instance is dehydratable for an amount of time longer than this threshold, it dehydrates the instance.
Idle Orchestrations	Number of idle orchestration instances currently hosted by the host instance. This refers to orchestrations that are not making progress but are not dehydratable, such as when the orchestration is blocked waiting for a receive, listen, or delay in an atomic transaction.
Message Box Database Connection Failures	Number of attempted database connections that failed since the host instance started.
Orchestrations Completed Per Sec	Average number of orchestrations completed per second.
Orchestrations Created Per Sec	Average number of orchestration instances created per second.
Orchestrations Dehydrated Per Sec	Average number of orchestration instances dehydrated per second.
Orchestrations Discarded Per Sec	Average number of orchestration instances discarded per second from memory. An orchestration can be discarded if the engine fails to persist its state.
Orchestrations Rehydrated Per Sec	Number of orchestration instances rehydrated per second.
Orchestrations Resident in Memory	Number of orchestration instances currently hosted by the host interface.
Orchestrations Scheduled for Dehydration	Number of dehydratable orchestrations for which there is a dehydration request pending.
Orchestrations Suspended Per Sec	Average number of orchestrations suspended per second.
Pending Messages	Number of received messages for which receipt has not yet been acknowledged to the message box.
Pending Work Items	Number of code execution blocks scheduled for execution.
Persistence Points Per Sec	Average number of orchestration instances persisted per second.
Runnable Orchestrations	Number of orchestration instances ready to execute.
Running Orchestrations	Number of orchestration instances currently executing.

7.14 Physical Memory and Application Domain Metrics

The metrics in this category provide information about loaded application domains and memory usage.

Table 7–14 Physical Memory and Application Domain Metrics

Metric	Description and User Action
Active Application Domains	Number of loaded orchestration application domains in the process. Much like the process, the application domain is designed as a security boundary that confines errors and faults to a specific domain. The application domain is designed as a virtual process that isolates applications. Often, especially for security reasons, you cannot avoid using multiple application domains. However, doing so can limit performance at startup. You can reduce the impact of multiple application domains by loading assemblies as domain neutral, enforcing efficient cross-AppDomain communication, using NeutralResourcesLanguageAttribute, and using serialization wisely.
Average Batch Factor	Number of persistence points reached since the host instance started, divided by the number of underlying transactions. This metric is beneficial where the Orchestration engine merges multiple atomic transactions into a single transaction. Assuming that a "persistence point" is really a segment boundary, this metric provides some substantial facts about the effect of batching atomic transactions. The greater the number, the greater the effect of "transaction batching" (that is, the less underlying transactions are being created).
Percent Used Physical Memory	Percentage of total used physical memory on the computer. The value of this metric should be about 75%. A consistently high value of 100% may indicate a problem.

7.15 Process Metrics

The metrics in this category provide process-related information.

Table 7–15 Process Metrics

Metric	Description and User Action
Page File Bytes	Current amount of virtual memory in bytes that this process has reserved for use in the paging file(s). Paging files store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory. If there is no paging file, this metric reflects the current amount of virtual memory that the process has reserved for use in physical memory.
Page File Bytes Peak	Maximum amount of virtual memory in bytes that this process has reserved for use in the paging file(s). Paging files store pages of memory used by the process that are not contained in other files.
Percent Processor Time	Elapsed time that all process threads used the processor to execute instructions. The average value of this metric should be approximately 70%. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count.
Private Bytes	Current size in bytes of memory that this process has allocated that cannot be shared with other processes.

Table 7–15 (Cont.) Process Metrics

Metric	Description and User Action
Thread Count	Number of threads currently active in this process. An instruction is the basic unit of execution in a computer, and a thread is the object that executes instructions. Every running process has at least one thread.
Virtual Bytes	Current size in bytes of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries.
Working Set	Current size in bytes of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they are soft-faulted back into the Working Set before leaving main memory. This metric measures the number of memory pages that each process uses. If the system has sufficient memory, it can maintain enough space in the working set so that IIS 6.0 rarely must perform disk operations. One indicator of memory sufficiency is how much the size of the process working set fluctuates in response to general memory availability on the server. Significant fluctuation can indicate a lack of available memory.

7.16 Transaction Metrics

The metrics in this category provide information about database transactions.

Table 7–16 Transaction Metrics

Metric	Description
Database Transactions Per Sec	Average number of database transactions performed per second.
Transactional Scopes Aborted	Number of aborted long-running or atomic scopes since the host instance started.
Transactional Scopes Committed Per Sec	Average number of transactions committed per second.
Transactional Scopes Compensated Per Sec	Average number of long-running or atomic scopes per second that have successfully completed compensation scopes.

8

Microsoft Commerce Server Metrics

The Microsoft Commerce Server 2002 Enterprise Edition management plug-in for Oracle Enterprise Manager provides hardware and software performance monitoring of Commerce Server using performance counters and application events. These counters can help you find hardware and software bottlenecks that may be restricting the flow of data.

This chapter provides descriptions for all Microsoft Commerce Server metric categories, and tables list and describe associated metrics for each category. The tables also provide user actions if any of the metrics for a particular category support user actions.

8.1 Active Server Pages Metrics

The metrics in this category provide information about the performance of the Internet Information Services (IIS) 6.0 Server using the performance counters described below. The counters are designed to monitor server performance and cannot be configured for individual sites.

Table 8–1 Active Server Pages Metrics

Metric	Description and User Action
Errors Per Sec	Average number of errors that occurred per second.
Request Execution Time	Number of milliseconds required to execute the most recent request. The value of this counter should be very close to zero. If it is not, check the code and try to determine the cause of the bottleneck that is consuming excessive execution time.
Request Wait Time	Number of milliseconds that the most recent request waited in the queue. This value should be very close to zero (less than 100 milliseconds), because this is the amount of time a request waits in the queue before it begins processing. You do not want users to experience long wait times to process their checkout transactions. If the value of this counter is high, check the code and try to determine the cause of the bottleneck that is consuming excessive execution time.
Requests Executing	Number of requests currently executing. IIS 6.0 has many threads that can simultaneously process requests. This value should be stable. Experience will help you set a threshold for a particular site.
Requests Per Sec	Rate that Active Server Pages (ASP) are processing requests. This includes both successful and failed requests. This counter provides and indication of the usage of the application. If this value is high, you may need to upgrade the hardware to handle the additional load.

Table 8–1 (Cont.) Active Server Pages Metrics

Metric	Description and User Action
Requests Queued	Requests waiting to be executed in the queue. There should not be a significant queue except at peak periods. This counter should not have a high value. It should be less than the Request Queue limit.
Requests Succeeded	Number of successfully fulfilled requests. A high value indicates a healthy system.
Sessions Total	Number of sessions that have run since the service was started.

8.2 Authentication Filter Metrics

The metrics in this category provide information about the checks that are initiated for every request received by the server using the performance counters described below.

Table 8–2 Authentication Filter Metrics

Metric	Description and User Action
AuthFilter: Failed NT Authentication Checks Per Sec	Number of failed Windows authentication checks per second. This counter is for a specific instance on the server. The value of this counter should be a nominal value and should be comparatively less than the successful authentication checks. A higher value indicates a possible credential problem with the system.
AuthFilter: Successful NT Authentication Checks Per Sec	Number of successful Windows authentication checks per second. This counter is for a specific instance on the server. The value of this counter should be high. Very low values may be of concern in cases when the traffic on the website is high, but the value of this counter is still nearly zero. Along with this counter, watch the failed authentication counter, whose value should be less than the value of this counter.
AuthFilter: Token Cache Entries Active	Current number of active token-cache entries in Windows authentication mode. This counter is for a specific instance on the server, and is reset to zero on an IISRESET or server reboot.
AuthFilter: Token Cache Entries Total	Total number of token-cache entries in Windows authentication mode. If the value of this counter is very high, increase the password cache size. It is located in the Global resource under the CS Authentication resource for the site, and is named Password-Cache Size. The default size is 10,000 objects. Another option is to set the PsObjectCacheSize property in the connection string while initializing the profiling service. After making these changes, you will need to perform an IISRESET.
AuthFilter: Taken Cache Size Bytes	Size, in bytes, of token-cache in Windows authentication mode.
AuthFilter: Total Failed NT Authentication Checks	Total number of failed Windows authentication checks.
AuthFilter: Total Requests Failed	Number of failures caused by a full filter-cache, or because an error occurred and a filter redirected to an error page.
AuthFilter: Total Successful NT Authentication Checks	Total number of successful Windows authentication checks by the authentication Internet Server Application Program Interface (ISAPI) filter. This is the total number of times the filter passed the credentials to Internet Information Services (IIS) 6.0 and returned successfully.
AuthFilter: Windows Authentication Cache Hits	Total number of user ID and password cache hits for an instance of a Windows authentication mode filter. This counter is for a specific instance on the server, and is reset to zero on an IISRESET or server reboot.

Table 8–2 (Cont.) Authentication Filter Metrics

Metric	Description and User Action
AuthFilter: Windows Authentication Cache Hits Per Sec	Number of user ID and password cache hits per second for an instance of the Windows authentication mode filter. This counter is for a specific instance on the server. You should monitor the cache hit rate and cache miss rate counters. If the cache miss rate is high in comparison to the cache hit rate, you should increase the size of your cache using the connection string parameter.
AuthFilter: Windows Authentication Cache Misses	Total number of user ID and password cache misses for an instance of a Windows authentication mode filter. A cache hit always follows a cache miss. This counter is for a specific instance on the server, and is reset to zero after an IISRESET or server reboot.
AuthFilter: Windows Authentication Cache Misses Per Sec	Number of user ID and password cache misses per second for an instance of a Windows authentication mode filter. This counter is for a specific instance on the server.

8.3 Authentication Manager Metrics

The metrics in this category provide information about the Authentication Manager, which is called from within ASP pages using the performance counters described below.

Table 8–3 Authentication Manager Metrics

Metric	Description and User Action
AuthMgr: Authentication Objects Per Sec	Number of Authentication Manager objects created per second at a particular site. A higher value f or this counter indicates a higher server load.
AuthMgr: Authentication Checks Failed Per Sec	Number of failed authentication checks per second at a particular site. The value of this counter should be a nominal value and should be comparatively less than the successful authentication checks. A higher value indicates a possible credential problem with the system.
AuthMgr: Authentication Checks Succeeded Per Sec	Number of successful authentication checks per second at a particular site. The value of this counter should be high. Very low values may be of concern in cases when the traffic on the website is high, but the value of this counter is still nearly zero. Along with this counter, watch the failed authentication counter, the value of which should be less than the value of this counter.
AuthMgr: Authentication Tickets Per Sec	Number of authenticated tickets set per second at a particular site.
AuthMgr: Custom Properties Per Sec	Number of custom properties set per second at a particular site.
AuthMgr: Get_Custom Properties Per Sec	Number of `Get_Custom_Properties` set per second at a particular site. The `GetProperty()` counter gets the custom property values from the user cookie.
AuthMgr: Total Authentication Checks Failed	Total number of failed authentication checks for a particular site.
AuthMgr: Total Authentication Checks Succeeded	Total number of successful authentication checks for a particular site.
AuthMgr: Total Authentication Objects Created	Total number of Authentication Manager objects created at a particular site.

Table 8–3 (Cont.) Authentication Manager Metrics

Metric	Description and User Action
AuthMgr: Total Authentication Tickets	Total number of authenticated tickets at a particular site.
AuthMgr: Total Custom Properties	Total number of custom properties set at particular site.
AuthMgr: Total Get_ Custom Properties	Total number of GetCustom-properties set at a particular site.

8.4 Commerce Server Response Metrics

The metric in this category provides information about the status of the Commerce Server.

Table 8–4 Commerce Server Response Metrics

Metric	Description
Status	A value of 1 or 0 indicates that the status is up or down based on the status of DMLService, List Manager, Predictor Service, and the WWW Service as ‘Running’ or ‘Stopped’.

8.5 Commerce Server Error Events Metrics

The metrics in this category provide information about error events generated by Commerce Server in the application log.

Table 8–5 Commerce Server Error Events Metrics

Metric	Description
Description	Description of the error event.
Windows Event Severity	Severity of the event.

8.6 Commerce Server Warning Events Metrics

The metrics in this category provide information about warning events generated by Commerce Server in the application log.

Table 8–6 Commerce Server Warning Events Metrics

Metric	Description
Description	Description of the warning event.
Windows Event Severity	Severity of the event.

8.7 Data Warehouse and Analysis Metrics

The metrics in this category provide information about Data Warehouse- and Analysis-related counters. Data Warehouse performance counters are used only when importing data groups, such as DTS tasks, users, and catalogs. Therefore, the Data Warehouse performance counters need to be set up as anonymous servers.

Table 8–7 Data Warehouse and Analysis Metrics

Metric	Description
Catalog Import: Total Catalogs	Total number of catalogs imported during this DTS import task.
Catalog Import: Total Categories	Total number of categories imported during this DTS import task.
Catalog Import: Total Products	Total number of products imported during this DTS import task.

8.8 Direct Mailer Metrics

The metrics in this category provide information about the Direct Mailer and List Manager system using the performance counters described below. The Direct Mailer uses a high-performance mail generation engine to create personalized messages using the lists that you manage with List Manager. A List Manager is used to import and manage mailing lists.

Table 8–8 Direct Mailer Metrics

Metric	Description and User Action
DML: Direct Mail Errors Per Sec	Number of errors returned by Direct Mailer per second. The value for this counter should be very low.
DML: Direct Mail Jobs Running	Total number of Direct Mailer jobs currently running for a particular instance. A higher value for this counter indicates a heavy server load.
DML: Mail Messages Sent Per Sec	Number of mail messages sent by all Direct Mailer jobs per second.
DML: Total Direct Mail Errors	Total number of errors returned by all Direct Mailer jobs.
DML: Total Messages Sent	Total number of mail messages sent by all Direct Mailer jobs.

8.9 Expression Evaluator Engine Metrics

The metrics in this category provide information about the performance of EEE using the performance counters described below. The EEE consists of the evaluation core (logic), the ExpressionStore, and the ExpressionEval objects. The EEE contains an Expression Cache that caches loaded expressions, object properties, and expression results.

Table 8–9 Expression Evaluator Engine Metrics

Metric	Description and User Action
EEE: Evaluations Per Sec	Number of expressions evaluated per second. **Note:** Displaying an advertisement or discount can cause more than one evaluation.
EEE: Expression Cache Size	Total number of expressions currently in the expression cache. The EEE engine has two internal caches: the Expression Cache and the Expression Result Cache. These EEE caches increase in size dynamically as items are added to these caches. Because these caches are dynamic, they are not customizable. To enhance the performance of the expression evaluator, it is recommended that you preload expressions into the expression cache using the LoadExpr or LoadAll methods of the ExpressionEval object. To avoid runtime expression load operations, it is recommended that you call the loadAll method in the site `Global.asa` file.
EEE: Property Cache Hits	Total number of property cache hits since the server was started. In addition to the Expression Cache, Commerce Server 2002 also includes an Expression Results Cache and an Expression Properties Cache. These caches grow or shrink dynamically as items are loaded or removed from them.
EEE: Property Reads Per Second	Total number of properties read by the Expression Evaluator since the server started.
EEE: Total Evaluation Errors	Total number of evaluation errors returned.

8.10 Marketing and Catalog Metrics

Commerce Server includes a Product Catalog System to manage products, create multilingual and multicurrency catalogs, and provide different search methods to quickly find needed products and services.

The metrics in this category provide information about the performance of the Product Catalog system using the performance counters described below.

Table 8–10 Marketing and Catalog Metrics

Metric	Description and User Action
Catalog Queries Per Sec	Number of queries made to the Product Catalog System per second. Query and FreeTextSearch record the values for the performance monitor counters. No other methods record the counters. Commerce Server 2002 implemented the same usage profile for both base catalogs and indexed view virtual catalogs to show the activities of the Product Catalog System.
LRU Cache Flushes Per Sec	Number of items flushed out of the cache per second to accommodate new items added to the cache. Records items flushed by the LRU algorithm and items flushed manually. The least recently used LRU cache counters indicate how well catalog caching is performing when catalog caching is used. An LRU cache that is performing well should have a low miss rate, a low flush rate, and a high hit rate.
LRU Cache Hits Per Sec	Number of cache hits per second for the cache. The least recently used LRU cache counters indicate how well catalog caching is performing when catalog caching is used. An LRU cache that is performing well should have a low miss rate, a low flush rate, and a high hit rate. Monitor the LRU cache counters, and tune your cache to achieve a 90 percent hit rate.
LRU Cache Misses Per Sec	Number of cache misses per second for the cache. The least recently used LRU cache counters indicate how well catalog caching is performing when catalog caching is used. An LRU cache that is performing well should have a low miss rate, a low flush rate, and a high hit rate.

Table 8–10 (Cont.) Marketing and Catalog Metrics

Metric	Description and User Action
LRU Cache Size	Total number of entries in the cache. Cache size can grow to 10 percent more than the size the user sets, at which time a flush occurs to return the cache size to the preset limit. LRU cache counters are grouped by site, not by the individual cache. The least recently used LRU cache counters indicate how well catalog caching is performing when catalog caching is used. An LRU cache that is performing well should have a low miss rate, a low flush rate, and a high hit rate.
Predictor Client Average Prediction Time	Average elapsed time required for a prediction to be returned.
Predictor Total Client Model Loads	Number of models loaded since the service was last restarted.
Predictor Total Client Predictions	Total number of predictions.

8.11 Memory Metrics

The metrics in this category provide information about the performance of memory-related parameters using the performance counters described below.

Table 8–11 Memory Metrics

Metric	Description
Available Bytes	Amount of physical memory, in bytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free, and zero page lists. The value should be greater than 4 Megabytes.
Committed Bytes	Amount of committed virtual memory, in bytes. Committed memory is the physical memory that has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average. The value of this counter should not be more than 75 percent of physical memory.
Page Faults Per Sec	Average number of pages faulted per second. It is measured in number of pages faulted per second because only one page is faulted in each fault operation; hence, this is also equal to the number of page fault operations. This counter includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory.) Most processors can handle large numbers of soft faults without significant consequences. However, hard faults, which require disk access, can cause significant delays.
Page Reads Per Sec	Rate at which the disk was read to resolve hard page faults. This metric shows the number of read operations without regard to the number of pages retrieved in each operation. Hard page faults occur when a process references a page in virtual memory that is not in a working set or elsewhere in physical memory, and must be retrieved from disk. The value should be less than one page per second. If the system is actually out of memory, this is the biggest indicator of the problem. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It includes read operations to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. Compare the value of Memory\\Pages Reads/sec to the value of Memory\\Pages Input/sec to determine the average number of pages read during each operation.

8.12 Network Metrics

The metrics in the Network category provide information about network performance using the performance counters described below.

Table 8–12 Network Metrics

Metric	Description and User Action
Bytes Received Per Sec	Rate, in seconds, at which bytes are received over this network adapter. The counted bytes include framing characters. This counter is a subset of Network Interface\Bytes Total/sec. If a network card approaches its maximum capacity, another should be added. If this value approaches the capacity of the network, then a higher bandwidth network might be necessary.
Bytes Sent Per Sec	The rate, in seconds, at which bytes are sent over this network adapter. The counted bytes include framing characters. This counter is a subset of Network Interface\Bytes Total/sec. If a network card approaches its maximum capacity, another should be added. If this value approaches the capacity of the network, a higher bandwidth network might be necessary.
Bytes Total Per Sec	Rate at which bytes are sent and received over each network adapter, including framing characters. Network Interface\\Bytes Received/sec is a sum of Network Interface\\Bytes Received/sec and Network Interface\\Bytes Sent/sec. If a network card approaches its maximum capacity, another should be added. If this value approaches the capacity of the network, a higher bandwidth network might be necessary.
Current Bandwidth	Estimate of the current bandwidth of the network interface in bits per second (BPS). For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this value is the nominal bandwidth.
Output Queue Length	Length of the output packet queue (in packets). Delays exist if this value is longer than two. Try to find and eliminate the bottleneck, if possible. Since the requests are queued by the Network Driver Interface Specification (NDIS) in this implementation, this will always be 0.
Packets Outbound Discarded	Number of outbound packets chosen to be discarded even though no errors had been detected to prevent transmission. One possible reason for discarding packets could be to free up buffer space.
Packets Outbound Errors	Number of outbound packets that could not be transmitted because of errors.
Packets Received Discarded	Number of outbound packets chosen to be discarded even though no errors had been detected that prevented transmission. One possible reason for discarding packets could be to free up buffer space.
Packets Received Errors	Number of inbound packets containing errors that prevented them from being delivered to a higher-layer protocol.

8.13 Physical Disk Metrics

The metrics in the Physical Disk category provide information about disk performance using the performance counters described below.

Table 8–13 Physical Disk Metrics

Metric	Description and User Action
Average Disk Queue Length	Average of disk queue length. If the disk is not fast enough to keep up with read and write requests, requests will queue up. The acceptable queue length is a function of the number of spindles in the array. Other counters that can be used to observe disk traffic include Physical Disk: Disk Reads/second and Physical Disk: Disk Writes/second. If necessary, consider adding more physical drives, such as a Redundant Array of Inexpensive Disks (RAID) system, to increase the number of spindles that can read and write, as well as to increase data transfer rates.
Disk Reads Per Sec	Number of disk reads per second on the physical disk. This counter should be well under the maximum capacity for the disk device. To enable this counter, run `diskperf -y` from the command shell and reboot the computer.
Disk Writes Per Sec	Number of disk writes per second on the physical disk. This counter should be well under the maximum capacity for the disk device. To enable this counter, run `diskperf -y` from the command shell and reboot the computer.
Percent Disk Time	Percentage of elapsed time that the selected disk drive is busy servicing read or write requests.

8.14 Pipelines Metrics

A pipeline is an extensible software framework that defines and links together one or more stages of a business process, running them in sequence to complete a specific task. The metrics in this category provide information about Pipeline performance using the performance counters described below.

Table 8–14 Pipelines Metrics

Metric	Description and User Action
Average Execution Time	Average execution time in microseconds for t he particular component of the pipeline. The execution time depends on the Service Level agreement with the client. It should not be high.
Errors Per Sec	Number of errors generated by the particular pipeline component per second.
Errors Total	The total number of Error Level 3 failures returned by the particular pipeline component (PIPEERRORLEV_FAIL) or FAILED HRESULT.
Warnings Per Sec	Number of warnings returned by the pipeline component per second.
Warnings Total	The total number of Error Level 2 warnings returned by the particular pipeline component (PIPEERRORLEV_WARN).

8.15 Process Metrics

Table 8–15 Process Metrics

Metric	Description and User Action
Creating Process ID	Process ID of the process that created the process. The creating process may have terminated, so this value may no longer identify a running process.
Handle Count	Total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in the process.
IO Data Bytes Per Sec	Rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.

Table 8–15 (Cont.) Process Metrics

Metric	Description and User Action
IO Data Operations Per Sec	Rate at which the process is issuing read and write I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
Percent Processor Time	Percentage of elapsed time that all process threads spend using the processors. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count. The average value of this counter should be approximately 70%.
Private Bytes	Current size, in bytes, of memory that this process has allocated that cannot be shared with other processes. Memory leaks are identified by a consistent and prolonged increase in Private Bytes. This is the best performance counter for detecting memory leaks. Values greater than 60% of total physical RAM begin to impact performance, especially during application and process restarts. An optimum value for this counter is a value whichever is minimum out of 60% of physical RAM and 800 MB.
Thread Count	Number of threads currently active in this process. An instruction is t he basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread. Thread count often increases when the load is too high. Its optimum value is expressed in the formula: `75 + ((maxWorkerThread + maxIoThreads) * #CPUs)`
Virtual Bytes	Current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries. The value of this counter should be 600 MB less than the size of the virtual address space; either 1.4 or 2.4 GB.
Virtual Bytes Peak	Maximum size, in bytes, of virtual address space the process has used at any one time. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. However, virtual space is finite, and the process might limit its ability to load libraries. The value of this counter should be 600 MB less than the size of the virtual address space; either 1.4 or 2.4 GB.
Working Set	Current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they are then soft-faulted back into the Working Set before leaving main memory. This counter measures the number of memory pages that each process uses. If the system has sufficient memory, it can maintain enough space in the working set so that IIS 6.0 rarely must perform disk operations. One indicator of memory sufficiency is how much the size of the process working set fluctuates in response to general memory availability on the server. Significant fluctuation can indicate a lack of available memory.

8.16 Processor Metrics

The metric in this category provides information about processor performance.

Table 8–16 Processor Metrics

Metric	Description and User Action
Percent Processor Time	Percentage of elapsed time that all process threads used the processor to execute instructions. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count. The value of this counter provides an indication about the processor time utilized by the applications running on the server. A consistently very high value for this counter indicates possible problems in the code and may require some refactoring of the application.

8.17 SQL Server Metrics

The metric in this category provides information about SQL server performance.

Table 8–17 SQL Server Metrics

Metric	Description
Transactions Per Sec	Number of transactions started for the database. A transaction is any activity that exchanges data. This counter indicates how much activity the SQL Server actually performs.

8.18 SQL Server Statistics Metrics

The metrics in this category provide information about SQL Server performance using the performance counters described below.

Table 8–18 SQL Server Statistics Metrics

Metric	Description and User Action
SQL Compilations Per Sec	Indicates the efficiency of the queries that the SQL Server is running. Reduce this number to reduce the CPU load on the SQL Server.
SQL Recompilations Per Sec	Indicates the efficiency of the queries that the SQL Server is running. Reduce this number to reduce the CPU load on the SQL Server.

8.19 System Metrics

The metrics in the System category provide information about system performance using the performance counters described below.

Table 8–19 System Metrics

Metric	Description and User Action
Context Switches Per Sec	Combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an executive or subsystem service. It is the sum of Thread\\Context Switches/sec for all threads running on all processors in the computer and is measured in numbers of switches. There are context switch counters on the System and Thread objects. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval. This counter shows the activities of the Product Catalog System. Heavy catalog queries can overload both the front-end Web server and the back-end catalog server. You can use the Commerce Server 2002 least recently used (LRU) cache to improve the performance of an overloaded Commerce Server 2002 Product Catalog System.
Processor Queue Length	Number of threads in the processor queue. Unlike the disk counters, this counter shows ready threads only, not threads that are running. A sustained processor queue of less than 10 threads per processor is normally acceptable, depending on the workload. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload.

8.20 User Profile Management Metrics

The User Profile Management (UPM) system provides an easy-to-use User Management and Personalization API that you can use to build Commerce Server applications. The metrics in this category provide information about UPM performance using the performance counters described below.

Table 8–20 User Profile Management Metrics

Metric	Description and User Action
UPM: Active Connection	Total number of currently active connections in the connection pool. This counter is for a server, not an instance.
UPM: Active Heap Count	Total number of currently active heaps in the heap pool. This counter is for a server, not an instance.
UPM: Cache Hit Rate	Number of cache hits per second for any SQL Server-based properties. This counter is for a server, not an instance. This counter is for a server, not an instance. If your cache miss rate is high in comparison to the cache hit rate, you should increase the size of your cache using the PsObjectCacheSize parameter in the connection string.
UPM: Cache Miss Rate	Number of cache misses per second for any SQL Server-based properties. This counter is for a server, not an instance. This counter is for a server, not an instance. If your cache miss rate is high in comparison to the cache hit rate, you should increase the size of your cache using the PsObjectCacheSize parameter in the connection string.
UPM: Cache Objects	Total number of cache objects. This counter is for a server, not an instance. This value should always be equal to or greater than the number of current users on the system. As the rate of this counter against time increases, you need to monitor the disk queue on the back-end data store.
UPM: Create Profile Latency	Cumulative latency for all profile object creations in one second. This counter is for the server, not an instance. The value of this counter should not be high, because a high value slows user authentication, even when the user already exists on the domain but is not replicated to all domain controllers.

Table 8–20 (Cont.) User Profile Management Metrics

Metric	Description and User Action
UPM: Delete Profile Latency	Cumulative latency for all profile object deletions in one second. This counter is for the server, not an instance.
UPM: Get Profile Latency	Cumulative latency for all profile object retrievals in one second. This counter is for the server, not an instance. The value of this counter should not be high, because a high value slows user authentication, even when the user already exists on the domain but is not replicated to all domain controllers.
UPM: Modify Profile Latency	Cumulative latency for all profile object updates in one second. This counter is for the server, not an instance.
UPM: Object Creations Per Sec	Number of profile objects created per second. This counter is for the server, not an instance.
UPM: Object Deletes Per Sec	Number of profile objects deleted per second. This counter is for the server, not an instance.
UPM: Object Modifies Per Sec	Number of profile objects modified per second. This counter is for the server, not an instance.
UPM: Object Reads Per Sec	Number of profile objects retrieved per second. This counter is for the server, not an instance.

8.21 Web Service Metrics

The metrics in the Web Service Metrics category provide information about Web service performance using the performance counters described below.

Table 8–21 Web Service Metrics

Metric	Description
CGI Requests	Current number of CGI requests being processed simultaneously by the WWW service.
Current Connections	Current number of connections established with the HTTP service. A threshold for this counter is dependent on many variables, such as the type of requests (ISAPI, CGI, static HTML, CPU utilization, and so on).
Get Requests	Current number of HTTP requests using the GET method made to the WWW service.
Head Requests	Current number of HTTP requests using the HEAD method made to the WWW service.
Logon Attempts Per Sec	
Post Requests	Current number of HTTP requests using the POST method made to the WWW service.
Total CGI Requests	Total number of all CGI requests that have been made since WWW service startup.
Total Connection Attempts (All Instances)	Total number of attempted connections to the WWW service since service startup.
Total Get Requests	Total number of HTTP requests that were using the GET method since WWW service startup.
Total Head Requests	Total number of HTTP requests that were using the HEAD method since WWW service startup.
Total Post Requests	Total number of HTTP requests that were using the POST method since WWW service startup.

9 Microsoft Exchange Server Metrics

Microsoft Exchange Server is a messaging and collaborative software product developed by Microsoft. It is commonly used by organizations and businesses that need to manage and store emails for multiple people. Enterprise Manager Grid Control allows you to discover and monitor Microsoft Exchange Servers that are in your environment

This chapter provides descriptions for all Microsoft Exchange Server metric categories, and tables list and describe associated metrics for each category.

9.1 DSAccess Domain Controllers Reachability Metrics

This category provides reachability metrics for DSAccess Domain Controllers.

Default Collection Interval — Every 5 minutes

Table 9–1 DSAccess Domain Controllers Reachability Metrics

Metric	Description
IsFast	When True, shows the domain controller response time has been less than two seconds.
IsInSync	Shows if the domain controller is synchronized with the Global Catalog server and with the Configuration domain controller.
IsUp	Shows whether the domain controller was available the last time Exchange attempted to access it.

9.2 Exchange - Cluster Resource State Metrics

This category provides metrics about Microsoft Exchange Cluster resource.

Default Collection Interval — Every 5 minutes

Table 9–2 Exchange - Cluster Resource State Metrics

Metric	Description
State	Shows the current state of the cluster resource.

9.3 Exchange - Connectors Status Metrics

This category provides connectors status metrics.

Default Collection Interval — Every 5 minutes

Table 9–3 Exchange - Connectors Status Metrics

Metric	Description
GroupDN	Shows the distinguished Name (DN) in Active Directory of the routing group.
Is Up	Shows that the Exchange Connector is operating normally.
Name	Shows the name of the Exchange Connector.

9.4 Exchange - Directory Service Flags Metrics

This category provides metrics related to directory service flags.

Default Collection Interval — Every 10 minutes

Table 9–4 Exchange - Directory Service Flags Metrics

Metric	Description
Critical-Data Flag	Shows 1 if the Domain Controller contains the Exchange server account, 0 otherwise.
GC-Capable Flag	Shows 1 if the Domain Controller is also a Global Catalog Server, 0 otherwise.
Is-Synchronized Flag	Shows 1 if the Domain Controller is synchronized.
Netlogon Flag	Shows 1 when DsGetDcName to the DC succeeds, indicating that NetLogon service is running, 0 otherwise.
OS Version Flag	Shows 1 when the DC satisfies DSAccess OS/Service Pack requirements, 0 otherwise.
SACL-Right Flag	Shows 1 if the Exchange server has a SeSecurityPrivilege on the Domain Controller, 0 otherwise.

9.5 Exchange - Directory Service Performance Metrics

This category provides directory service performance metrics.

Default Collection Interval — Every 10 minutes

Table 9–5 Exchange - Directory Service Performance Metrics

Metric	Description
LDAP Read Time(in ms)	Shows the time (in ms) taken to send an LDAP read request to the specified Domain Controller and receive a response.
LDAP Read calls Per Second	Shows the number of Depth 0 LDAP search calls made by this process per second.
LDAP Search Calls Per Second	Shows the number of Depth 1 or 2 LDAP Search calls made by this process per second.
LDAP Search Time(in ms)	Shows the time (in ms) taken to send an LDAP search request and receive a response.

9.6 Exchange - Links Performance Metrics

This category provides metrics links performance metrics.

Default Collection Interval — Every 10 minutes

Table 9–6 Exchange - Links Performance Metrics

Metric	Description
Global Stop	Shows whether the virtual server for the link is in disabled mode. If True, the virtual server is disabled.
Increasing-Time(milliseconds)	Shows the time (in milliseconds) that the number of messages waiting to be transferred by the link has been increasing.
Number Of Messages	Shows the number of messages that are waiting for transmission across the link.
Oldest Message	Shows the date and time when the oldest message, which is still waiting to be transmitted, was received into the link.
Size Of Queue(Bytes)	Shows the total size of the messages in the link (in bytes).
Size Of Queue(MB)	Shows the total size of the messages in the link (in megabytes).
State Active	Shows if the link is active. If True, then it is active.
State Frozen	Shows whether the link is currently frozen.
State Ready	Shows if the link is ready to accept new messages. If True, then is it ready.
State Remote	Shows if the destination for messages in this link is on a remote server.
State Retry	Shows if the link is retrying a transmission that was unsuccessful. If True, then it is retrying.
State Scheduled	When True, shows that the link is scheduled for periodic activation, as compared with asynchronous, on-demand activation.

9.7 Exchange - Server State Metrics

This category provides information about the host running Microsoft Exchange Server.

Default Collection Interval — Every 5 minutes

Table 9–7 Exchange - Server State Metrics

Metric	Description
DN	Shows the Microsoft Active Directory distinguished name (DN) of the Exchange server object.
GUID	Shows the GUID of the Exchange Server object in Active Directory.
Group DN	Shows the DN of the Exchange Server routing group in Active Directory.
Group GUID	Shows the globally unique identifier (GUID) of the Exchange Server routing group in Active Directory.
Server Maintenance	Shows if the notifications set up in the Exchange Server System Manager Monitoring and Status page have been disabled.
Server State	Shows the current status of Microsoft Exchange Server.
Server State String	Shows the current state of the computer running Exchange Server.
Services State	Shows the current state of the monitoring services running on the Exchange Server host.
Services State String	Shows the current state of the monitoring services running on the Exchange Server host.

9.8 Exchange - Top Sized Mailboxes Metrics

This category provides metrics about the top sized mailboxes.

Default Collection Interval — Every 60 minutes

Table 9–8 Exchange - Top Sized Metrics

Metric	Description
Mailbox Display Name	Shows the name of the Mailbox.
Size (MB)	Shows the cumulative size of all of the messages in the Mailbox (in megabytes).
Size (bytes)	Shows the cumulative size of all of the messages in the Mailbox (in bytes).

9.9 Exchange - Top Sized Public Folders Metrics

This category provides metrics about the top sized public folders in the exchange server.

Default Collection Interval — Every 60 minutes

Table 9–9 Exchange - Top Sized Public Folders Metrics

Metric	Description
Name	Shows the name of the Public Folder.
Total Message Size	Shows the total size of the folder (in bytes).
Total Message Size (MB)	Shows the total size of the folder (in megabytes).

9.10 Exchange Core Processes Performance Metrics

This category provides exchange core processes performance metrics.

Default Collection Interval — Every 10 minutes

Table 9–10 Exchange Core Processes Performance Metrics

Metric	Description
CPU utilization(%)	Shows the percentage of elapsed time that all of process threads used the processor to execute instructions.
Virtual Bytes (MB)	Shows the current size (in megabytes) of the virtual address space the process is using.
Working Set (MB)	Shows the current size (in megabytes) of the Working Set of this process.

9.11 Exchange File System Driver Status Metrics

This category provides metrics related to exchange file system driver status.

Default Collection Interval — Every 10 minutes

Table 9–11 Exchange File System Driver Status Metrics

Metric	Description
Status	Shows the current status of the object.

9.12 Exchange Host Metrics

This category provides metrics about the host on which Microsoft Exchange Server is running.

Default Collection Interval — Every 10 minutes

Table 9–12 Exchange Host Metrics

Metric	Description
CPU Utilization (%)	Shows the total CPU utilization (in percentage) by all the processes running on the host.
Memory Utilization	Shows the total memory utilization (in megabytes) of all processes.
Memory Utilization (%)	Shows the percentage of main memory used by the processes running on the host.
Total Physical Memory	Shows the total main memory installed on the host.

9.13 Exchange Services Metrics

This category provides metrics related to information on the performance of Exchange Services.

Default Collection Interval — Every 5 minutes

Table 9–13 Exchange Services Metrics

Metric	Description
Caption	Shows a short textual description (one-line string) of the Service.
Description	Shows the description of the Service.
PathName	Shows the fully qualified path to the service binary file that implements the service.
ProcessId	Shows the process identifier of the service.
State	Shows the current state of the base service.
Status	Shows the current status of the object.

9.14 Information Store General Metrics

This category provides information store general metrics.

Default Collection Interval — Every 10 minutes

Table 9–14 information Store General Metrics

Metric	Description
Active Connection Count	Shows the number of connections that have shown some activity in the last 10 minutes.
Active User Count	Shows the number of user connections that have shown some activity in the last 10 minutes.
Connection Count	Shows the number of client processes connected to the information store.
RPC Averaged Latency	Shows the RPC latency (in milliseconds) averaged for the past 1024 packets.
RPC Operations Per second	Shows the rate at which the RPC operations occur.
User Count	Shows the number of users connected to the information store.
Virus Scan Files Quarantined Per Second	Shows the rate at which separate files are put into quarantine by virus scanner.
Virus Scan Files Scanned Per Second	Shows the rate at which separate files are processed by virus scanner.
Virus Scan Queue Length	Shows the current number of outstanding requests that are queued for virus scanning.

9.15 Information Store Mailbox Stores Metrics

This category provides metrics related to information store mailbox stores.

Default Collection Interval — Every 10 minutes

Table 9–15 Information Store Mailbox Stores Metrics

Metric	Description
Average Delivery Time	Shows the average time (in milliseconds) between the submission to the mailbox store and the delivery to all local recipients for the last 10 messages.
Local Delivery Rate	Shows the rate at which messages are delivered locally.
Messages Delivered Per Second	Shows the rate at which messages are delivered to all recipients.
Messages Submitted Per Second	Shows the rate at which clients submit messages.
Receive Queue Size	Shows the number of messages in the mailbox store's receive queue.
Send Queue Size	Shows the number of messages in the mailbox store's send queue.

9.16 Information Store Public Stores Metrics

This category provides metrics related to information store public stores metrics.

Default Collection Interval — Every 10 minutes

Table 9–16 Information Store Public Stores Metrics

Metric	Description
Average Delivery Time(ms)	Shows the average time (in milliseconds) between the submission to the public store and the delivery to all local recipients for the last 10 messages.
Messages Delivered Per Minute	Shows the rate at which messages are delivered to all recipients.
Messages Delivered Per Second	Shows the rate at which messages are delivered to all recipients.
Messages Submitted Per Minute	Shows the rate at which messages are submitted by clients.
Messages Submitted Per Second	Shows the rate at which messages are submitted by clients.
Receive Queue Size	Shows the number of messages in the public store's receive queue.
Send Queue Size	Shows the number of messages in the public store's send queue.

9.17 Logical Disk Performance Metrics

This category provides logical disk performance metrics.

Default Collection Interval — Every 10 minutes

Table 9–17 Logical Disk Performance Metrics

Metric	Description
Disk Reads Per Second	Shows the number of disk reads per second.
Disk Size(MB)	Shows the size of the logical disk.
Disk Writes Per Second	Shows the number of disk writes per second.

Table 9–17 (Cont.) Logical Disk Performance Metrics

Metric	Description
Free Space (%)	Shows the percentage of free Space on the disk.
Free Space (MB)	Shows the amount of free space on the logical disk in MB.

9.18 MTA Connections Metrics

This category provides metrics related to MTA Connections.

Default Collection Interval — Every 10 minutes

Table 9–18 MTA Connections Metrics

Metric	Description
Connector Index	Shows the unique index associated with a connector entity.
Current Inbound Connections	Shows the current number of inbound (remote initiated) associations with the entity.
Current Outbound Connections	Shows the current number of outbound (locally initiated) associations with the entity.
Messages Received Per Second	Shows the rate at which the messages are received from the connected entity.
Messages Sent Per Second	Shows the rate at which the messages are sent to the connected entity.
Size of OutBound Messages Queue	Shows the number of outstanding messages queued for transfer to the entity.

9.19 Response Metrics

This category provides status metrics for the Microsoft Exchange Server.

Default Collection Interval — Every 5 minutes

Table 9–19 Response Metrics

Metric	Description
Status	Shows the current status of Microsoft Exchange Server. The status can be Agent Unreachable, Under Blackout, Metric Collection Error, Up, Down, Partially Up, Agent Down, or Status Pending.

9.20 SMTP Server Metrics

This category provides SMTP server metrics.

Default Collection Interval — Every 10 minutes

Table 9–20 SMTP Server Metrics

Metric	Description
Inbound Connections Current	Shows the total number of connections currently inbound.
Local Queue Length	Shows the number of messages in the local queue.
Messages Received Per Second	Shows the rate at which inbound messages are being received.
Messages Sent Per Second	Shows the rate at which outbound messages are being sent.

Table 9–20 (Cont.) SMTP Server Metrics

Metric	Description
Outbound Connections Current	Shows the number of connections currently outbound.
Remote Queue Length	Shows the number of messages in the remote queue.

9.21 Windows Event log - MS Exchange Error Events Metrics

This category provides metrics related to MS Exchange error events.

Default Collection Interval — Every 10 minutes

Table 9–21 Windows Event log - MS Exchange Error Events Metrics

Metric	Description
Description	Shows the event message as it appears in the NT Eventlog.
Record Number	Shows the event within the NT Eventlog logfile.
Windows Event Severity	Shows the severity of the windows event.

10

Microsoft Internet Information Services Metrics

This chapter provides descriptions for all Microsoft Internet Information Services (IIS) metric categories, and tables list and describe associated metrics for each category. The tables also provide user actions if any of the metrics for a particular category support user actions.

10.1 ASP Metrics

If you are running Active Server Pages (ASP) on your server, ASP metrics can help you determine how well the server or site is responding to ASP requests. ASP metrics monitor server performance; you cannot monitor individual ASP applications because ASP metrics collect global data across the entire WWW service.

Default Collection Interval — Every 15 minutes

Table 10–1 ASP Metrics

Metric	Description
Errors Per Sec	Number of errors generated per second during the execution of HTTP requests.
Errors Total	Total number of errors that occurred during the execution of HTTP requests. This includes parser, compilation, or run-time errors. This counter represents the sum of the Errors During Compilation, Errors During Preprocessing, and Errors During Execution counters. A well-functioning Web server should not generate errors.
Request Wait Time	Time consume by the request in the "wait" state.
Requests Disconnected	Number of requests that were disconnected because a communication failure occurred.
Requests Executing	Number of requests currently executing.
Requests Failed Total	Number of requests that failed because of errors, authorization failure, and rejections.
Requests Not Authorized	Number of requests that failed because access rights were insufficient.
Requests Not Found	Number of requests made for files that were not found.
Requests Per Sec	Number of requests that were executed per second.
Requests Queued	Number of requests in the queue waiting to be serviced. If this number increases as the number of client requests increases, the Web server has reached the limit of concurrent requests that it can process. The default maximum for this counter is 5,000 requests. You can change this setting in the computer's `Machine.config` file.
Requests Rejected	Total number of requests that were not executed because insufficient server resources existed to process them. This counter represents the number of requests that return a 503 HTTP status code, which indicates that the server is too busy.
Requests Succeeded	Number of requests that executed successfully.

Table 10–1 (Cont.) ASP Metrics

Metric	Description
Requests Timed Out	Number of requests that timed out.
Requests Total	Number of requests made since the service was started.
Session Duration	Length of time in milliseconds that the most recent session lasted.
Sessions Current	Number of sessions currently being serviced.
Sessions Timed Out	Number of sessions that have timed out.
Sessions Total	Number of sessions that have run since the service was started.
Transactions Per Sec	Average number of transactions per second that have been started.

10.2 ASP.Net Metrics

The metrics in this category provide information about all ASP.Net applications on a Web server computer.

Default Collection Interval — Every 15 minutes

Table 10–2 ASP.Net Metrics

Metric	Description and User Action
ASP.Net: Application Restarts	Number of times the applications restarts. An application can restart because changes were made to the `Web.config` file or to assemblies stored in the application's \Bin directory, or because too many changes occurred in Web Forms pages. Sudden increases in this counter can mean that your Web application is shutting down. If an unexpected increase occurs, be sure to investigate it promptly. This value resets every time IIS is restarted. Recreating the application domain and recompiling pages requires time. Therefore, unforeseen restarts should be investigated. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements for the adequate 'application restarts' value.
ASP.Net: Applications Running	Number of applications running on the server computer.
ASP.Net: Request Execution Time	Number of milliseconds required to execute the last request. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net: Request Wait Time	Number of milliseconds that the most recent request waited in the queue for processing. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net: Requests Current	Number of requests currently handled by the ASP.Net ISAPI. This includes those that are queued, executing, or waiting to be written to the client. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net: Requests Queued	Number of requests in the queue waiting to be serviced. If this number increases as the number of client requests increases, the Web server has reached the limit of concurrent requests that it can process. The default maximum for this counter is 5,000 requests. You can change this setting in the computer's `Machine.config` file. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

Table 10–2 (Cont.) ASP.Net Metrics

Metric	Description and User Action
ASP.Net: Requests Rejected	Total number of requests that were not executed because insufficient server resources existed to process them. This counter represents the number of requests that return a 503 HTTP status code, which indicates that the server is too busy. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net: Worker Process Restarts	Number of times that a worker process restarted on the server computer. A worker process can be restarted if it fails unexpectedly or when it is intentionally recycled. If worker process restarts increase unexpectedly, investigate immediately. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net: Worker Processes Running	Number of worker processes running on the server computer. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

10.3 ASP.Net Applications Metrics

The metrics in this category monitor the performance of a single instance of an ASP.Net application.

Default Collection Interval — Every 15 minutes

Table 10–3 ASP.Net Applications Metrics

Metric	Description and User Action
ASP.Net Apps: Anonymous Requests	Number of requests that use anonymous authentication.
ASP.Net Apps: Anonymous Requests/sec	Average number of requests made per second that use anonymous authentication.
ASP.Net Apps: Cache API Entries	Number of entries currently in the user cache.
ASP.Net Apps: Cache API Hit Ratio	Total hit-to-miss ratio of user cache requests.
ASP.Net Apps: Cache API Turnover Rate	Number of additions and removals to the user cache per second. A high turnover rate indicates that items are being quickly added and removed, which can be expensive. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net Apps: Cache Total Entries	Total number of entries in the cache. This counter includes both internal use of the cache by the ASP.Net framework and external use of the cache through exposed APIs.
ASP.Net Apps: Cache Total Hit Ratio	Ratio of cache hits to cache misses. This counter includes both internal use of the cache by ASP.Net and external use of the cache through exposed APIs.
ASP.Net Apps: Cache Total Hits	Total number of responses served from the cache. This counter includes both internal use of the cache by the ASP.Net framework and external use of the cache through exposed APIs.
ASP.Net Apps: Cache Total Turnover Rate	Number of additions to and removals from the cache per second. Use this counter to help determine how efficiently the cache is being used. If the turnover rate is high, the cache is not being used efficiently. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.

Table 10–3 (Cont.) ASP.Net Applications Metrics

Metric	Description and User Action
ASP.Net Apps: Compilations Total	Number of errors that occurred during dynamic compilation. Excludes parser and run-time errors.
ASP.Net Apps: Errors	Number of errors that occurred during parsing. Excludes compilation and run-time errors.
ASP.Net Apps: Output Cache Entries	Total number entries in the output cache. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net Apps: Output Cache Hit Ratio	Percentage of total requests that were serviced from the output cache.
ASP.Net Apps: Output Cache Turnover Rate	Average number of additions to and removals from the output cache per second. If the turnover is large, the cache is not being used effectively.
ASP.Net Apps: Pipeline Instance Count	Number of request pipeline instances that exist for this application. Since only one thread of execution can run within a pipeline instance, this number provides the maximum number of concurrent requests being processed for a given application. It is often better for this number to be low when under load, which signifies that the CPU is being well utilized.
ASP.Net Apps: Requests	Number of requests waiting to be processed. When this number starts to increment linearly with respect to client load, the limit of concurrent requests processed on the machine has been reached.
ASP.Net Apps: Requests Executing	Number of requests that are currently executing.
ASP.Net Apps: Requests Failed	Total number of failed requests. All status codes greater than or equal to 400 increment this counter. Requests that cause a 401 status code increment this counter and the Requests Not Authorized counter. Requests that cause a 404 or 414 status code increment this counter and the Requests Not Found counter. Requests that cause a 500 status code increment this counter and the Requests Timed Out counter.
ASP.Net Apps: Requests Not Authorized	Number of requests that failed because of a lack of authorization (status code 401).
ASP.Net Apps: Requests Not Found	Number of requests that failed because resources were not found (status codes 404 and 414).
ASP.Net Apps: Requests Per Sec	Current throughput of the application. Under constant load, this number should remain within a certain range, barring other server work (garbage collection, cache cleanup thread, external server tools, and so forth).
ASP.Net Apps: Requests Succeeded	Number of requests that executed successfully (status code 200).
ASP.Net Apps: Requests Timed Out	Number of requests that timed out.
ASP.Net Apps: Requests in Application Queue	Number of requests in the application request queue.
ASP.Net Apps: Total Errors	Total number of errors that occurred during the execution of HTTP requests, which includes parser, compilation, or run-time errors. This counter represents the sum of the Errors During Compilation, Errors During Preprocessing, and Errors During Execution counters. A well-functioning Web server should not generate errors.
ASP.Net Apps: Total Requests	Total number of requests made since the service started.
ASP.Net Apps: Total Requests Executing	Sum of all requests executing since the server started.
ASP.Net Apps: Total Requests Failed	Sum of all requests that failed since the server started.

Table 10–3 (Cont.) ASP.Net Applications Metrics

Metric	Description and User Action
ASP.Net Apps: Total Requests in Application Queue	Sum of requests in the application request queue since the server started.
ASP.Net Apps: Total Requests Not Authorized	Total number of requests that failed because of lack of authorization since the server started (status code 401).
ASP.Net Apps: Total Requests Not Found	Sum of requests that failed because resources were not found since the server started (status codes 404 and 414).
ASP.Net Apps: Total Requests Succeeded	Total number of requests that executed successfully since the server started (status code 200).
ASP.Net Apps: Total Requests Timed Out	Total number of requests that timed out since the server started (status code 500).

10.4 ASP.Net V1.1.4322 Metrics

The metrics in this category monitor the performance of a single instance of an ASP.Net 1.1 application. The 1.1 designation in the Metric column below indicates the version of the ASP .NET framework.

Default Collection Interval — Every 15 minutes

Table 10–4 ASP.Net V1.1.4322 Metrics

Metric	Description and User Action
ASP.Net 1.1: Application Restarts	Number of times the application restarts. An application can restart because changes were made to the `Web.config` file or to assemblies stored in the application's \Bin directory, or because too many changes occurred in Web Forms pages. Sudden increases in this counter can mean that your Web application is shutting down. If an unexpected increase occurs, be sure to investigate it promptly. This value resets every time IIS is restarted. Recreating the application domain and recompiling pages requires time; therefore, unforeseen restarts should be investigated. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ASP.Net 1.1: Applications Running	Number of applications that are running on the server computer.
ASP.Net 1.1: Request Execution Time	Number of milliseconds required to execute the last request. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ASP.Net 1.1: Request Wait Time	Number of milliseconds that the most recent request waited in the queue for processing. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ASP.Net 1.1: Requests Current	Number of requests currently handled by the ASP.Net ISAPI. This includes those that are queued, executing, or waiting to be written to the client. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.

Table 10–4 (Cont.) ASP.Net V1.1.4322 Metrics

Metric	Description and User Action
ASP.Net 1.1: Requests Queued	Number of requests in the queue waiting to be serviced. If this number increases as the number of client requests increases, the Web server has reached the limit of concurrent requests that it can process. The default maximum for this counter is 5,000 requests. You can change this setting in the computer's `Machine.config` file. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ASP.Net 1.1: Requests Rejected	Total number of requests that were not executed because of insufficient server resources to process them. This counter represents the number of requests that return a 503 HTTP status code, which indicates that the server is too busy. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ASP.Net 1.1: Worker Process Restarts	Number of times that a worker process restarted on the server computer. A worker process can be restarted if it fails unexpectedly or when it is intentionally recycled. If worker process restarts increase unexpectedly, investigate immediately. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ASP.Net 1.1: Worker Processes Running	Number of worker processes that are running on the server computer. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.

10.5 ASP.Net V1.1.4322 Applications Metrics

The metrics in this category monitor the performance of a single instance of an ASP.Net application. The 1.1 designation in the Metric column below indicates the version of ASP .NET.

Default Collection Interval — Every 15 minutes

Table 10–5 ASP.Net V1.1.4322 Applications Metrics

Metric	Description and User Action
ASP.Net 1.1 Apps: Anonymous Requests	Number of requests that use anonymous authentication.
ASP.Net 1.1 Apps: Anonymous Requests Per Sec	Average number of requests made per second that use anonymous authentication.
ASP.Net 1.1 Apps: Cache API Entries	Number of entries currently in the user cache.
ASP.Net 1.1 Apps: Cache API Hit Ratio	Total hit-to-miss ratio of user cache requests.
ASP.Net 1.1 Apps: Cache API Turnover Rate	Number of additions and removals to the user cache per second. A high turnover rate indicates that items are being quickly added and removed, which can be expensive. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net 1.1 Apps: Cache Total Entries	Total number of entries in the cache. This counter includes both internal use of the cache by the ASP.Net framework and external use of the cache through exposed APIs.
ASP.Net 1.1 Apps: Cache Total Hit Ratio	Ratio of cache hits to cache misses. This counter includes both internal use of the cache by ASP.Net and external use of the cache through exposed APIs.

Table 10–5 (Cont.) ASP.Net V1.1.4322 Applications Metrics

Metric	Description and User Action
ASP.Net 1.1 Apps: Cache Total Hits	Total number of responses served from the cache. This counter includes both internal use of the cache by ASP.Net and external use of the cache through exposed APIs.
ASP.Net 1.1 Apps: Cache Total Turnover Rate	Number of additions to and removals from the cache per second. Use this counter to help determine how efficiently the cache is being used. If the turnover rate is high, the cache is not being used efficiently. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net 1.1 Apps: Compilations Total	Number of errors that occurred during dynamic compilation. Excludes parser and run-time errors.
ASP.Net 1.1 Apps: Errors	Number of errors that occurred during parsing. Excludes compilation and run-time errors.
ASP.Net 1.1 Apps: Output Cache Entries	Total number of entries in the output cache. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
ASP.Net 1.1 Apps: Output Cache Hit Ratio	Percentage of total requests that were serviced from the output cache.
ASP.Net 1.1 Apps: Output Cache Turnover Rate	Average Number of additions to and removals from the output cache per second. If the turnover is high, the cache is not being used effectively.
ASP.Net 1.1 Apps: Pipeline Instance Count	This metric pertains to .NET version 1.1, and shows the number of request pipeline instances that exist for this application. Since only one thread of execution can run within a pipeline instance, this number provides the maximum number of concurrent requests being processed for a given application. It is often better for this number to be low when under load, which signifies that the CPU is being well utilized.
ASP.Net 1.1 Apps: Requests	This metric pertains to .NET version 1.1, and shows the number of requests waiting to be processed. When this number starts to increment linearly with respect to client load, the limit of concurrent requests processed on the machine has been reached.
ASP.Net 1.1 Apps: Requests Executing	Number of requests that are currently executing.
ASP.Net 1.1 Apps: Requests Failed	Total number of failed requests. All status codes greater than or equal to 400 increment this counter. Requests that cause a 401 status code increment this counter and the Requests Not Authorized counter. Requests that cause a 404 or 414 status code increment this counter and the Requests Not Found counter. Requests that cause a 500 status code increment this counter and the Requests Timed Out counter.
ASP.Net 1.1 Apps: Requests in Application Queue	Number of requests in the application request queue.
ASP.Net 1.1 Apps: Requests Not Auth	Number of requests that failed because of lack of authorization (status code 401).
ASP.Net 1.1 Apps: Requests Not Found	Number of requests that failed because resources were not found (status code 404 and 414).
ASP.Net 1.1 Apps: Requests Per Sec	This metric pertains to .NET version 1.1 and the current throughput of the application. Under constant load, this number should remain within a certain range, barring other server work (garbage collection, cache cleanup thread, external server tools, and so forth).
ASP.Net 1.1 Apps: Requests Succeeded	Number of requests that executed successfully (status code 200).
ASP.Net 1.1 Apps: Requests Timed Out	Number of requests that timed out (status code 500).

Table 10–5 (Cont.) ASP.Net V1.1.4322 Applications Metrics

Metric	Description and User Action
ASP.Net 1.1 Apps: Total Errors	Total number of errors that occurred during the execution of HTTP requests, which includes parser, compilation, or run-time errors. This counter represents the sum of the Errors During Compilation, Errors During Preprocessing, and Errors During Execution counters. A well-functioning Web server should not generate errors.
ASP.Net 1.1 Apps: Total Requests	Total number of requests made since the service started.
ASP.Net 1.1 Apps: Total Requests Executing	Sum of requests executing since the server started.
ASP.Net 1.1 Apps: Total Requests Failed	Sum of all requests that failed since the server started.
ASP.Net 1.1 Apps: Total Requests in Application Queue	Sum of requests in the application request queue since the server started.
ASP.Net 1.1 Apps: Total Requests Not Authorized	Total number of requests that failed because of a lack of authorization since the server started (status code 401).
ASP.Net 1.1 Apps: Total Requests Not Found	Sum of requests that failed because resources were not found since the server started (status codes 404 and 414).
ASP.Net 1.1 Apps: Total Requests Succeeded	Total number of requests that executed successfully since the server started (status code 200).
ASP.Net 1.1 Apps: Total Requests Timed Out	Total number of requests that timed out since the server started (status code 500).

10.6 FTP Service Metrics

The metrics in this category provide information about the FTP service (number of total files received, sent, and so forth).

Default Collection Interval — Every 15 minutes

Table 10–6 FTP Service Metrics

Metric	Description
Bytes Received Per Sec	Rate the FTP service receives data bytes.
Bytes Sent Per Sec	Rate the FTP service sends data bytes.
Bytes Total Per Sec	Sum of Bytes Sent Per Sec and Bytes Received Per Sec. This is the total rate of bytes the FTP service has transferred.
Current Anonymous Users	Number of users that currently have an anonymous connection using the FTP service.
Current Connections	Current number of connections established with the FTP service.
Current Non Anonymous Users	Number of users that currently have a non anonymous connection using the FTP service.
Maximum Connections	Maximum number of simultaneous connections established with the FTP service.
Total Connection Attempts (All Instances)	Number of connections attempted using the FTP service since service startup. This counter is for all instances listed.
Total Files Received	Total number of files the FTP service received since service startup.
Total Files Sent	Total number of files the FTP service sent since service startup.
Total Files Transferred	Sum of Total Files Sent and Total Files Received. This is the total number of files the FTP service has transferred since service startup.

10.7 FTP and WWW Service Error Events Metrics

The metrics in this category show the error events generated by the World Wide Web Publishing Service (WWW service) in the Windows event log file.

Default Collection Interval — Every 2 hours

Table 10–7 FTP and WWW Service Error Events Metrics

Metric	Description
Date-Time	Date and time when the error was generated.
Description	Description text of the error mentioned in the event log file.
Event ID	Every error generated has an event ID or record number. Using this event ID, you can search for more information on the web and MSN technet.
Log Name	Name of the log file where the error was generated.
Source	Component that generated the error or warning (that is, NTDS Inter-site messaging).
Windows Event Severity	Severity of the error.

10.8 FTP and WWW Service Warning Events Metrics

The metrics in this category provide the names of the various domain controllers in the forest and the role they are playing.

Default Collection Interval — Every 2 hours

Table 10–8 FTP and WWW Service Warning Events Metrics

Metric	Description
Date-Time	Date and time when the warning was generated.
Description	Description text of the warning mentioned in the event log file.
Event ID	Every warning generated has an event ID or record number. Using this event ID, you can search for more information on the web and MSN technet.
Log Name	Name of the log file where the warning was generated.
Source	Component that generated the warning (that is, NTDS Inter-site messaging).
Windows Event Severity	Severity of the warning.

10.9 IIS Global Service Metrics

Global Service metrics help to monitor FTP, SMTP, and NNTP services as a whole. The metrics in this category provide performance information for these services. If the service that you want to monitor (FTP, SMTP, or NNTP) is not installed or is not running, a zero value will be returned for the columns of these metrics.

Default Collection Interval — Every 15 minutes

Table 10–9 IIS Global Service Metrics

Metric	Description
Active Flushed Entries	Number of user-mode cache entries that have been flushed, though memory is still allocated for these entries. The allocated memory is released after all current transfers are complete.
BLOB Cache Hit Ratio	Ratio of BLOB cache hits to total cache requests.
BLOB Cache Hits Percent	Percentage of BLOB cache hits to total cache requests.
Current BLOBs Cached	BLOB information blocks currently in the cache.
Current File Cache Memory Usage	Number of bytes currently used for the user-mode file cache.
Current Files Cached	Number of files whose content is currently in the user-mode cache.
Current URIs Cached	Number of Uniform Resource Identifiers (URI) information blocks currently stored in the user-mode cache.
File Cache Flushes	Number of file cache flushes after server startup.
File Cache Hits Percent	Ratio of user-mode file cache hits to total cache requests made since the WWW service started up.
URI Cache Hits Percent	Ratio of URI cache hits to total cache requests made since the WWW service started up.

10.10 IIS Response Metrics

The metric in the IIS Response category provides information about the current status of the IIS server.

Default Collection Interval — Every minute

Table 10–10 IIS Response Metrics

Metric	Description
Status	If the value is 1, IIS is up. Otherwise, it is down.

10.11 IPV4 Transport Layer Metrics

The metrics in this category monitor IP datagrams, which are the units of data that IP sends down the protocol stack to the network interface, such as a network adapter.

Default Collection Interval — Every 15 minutes

Table 10–11 IPV4 Transport Layer Metrics

Metric	Description
IPV4: Datagrams Per Sec	Overall transmission rate for IP datagrams being sent and received over the network interfaces. This is the sum of Datagrams Sent Per Sec and Datagrams Received Per Sec.
IPV4: Datagrams Received Per Sec	Rate at which IP datagrams are received from the network interfaces. This counter does not include datagrams forwarded to another server.
IPV4: Datagrams Sent Per Sec	Rate at which IP datagrams are sent to the network interfaces. This counter does not include datagrams forwarded to another server.

10.12 IPV6 Transport Layer Metrics

The metrics in this category monitor IP datagrams, which are the units of data that IP sends down the protocol stack to the network interface, such as a network adapter.

Default Collection Interval — Every 15 minutes

Table 10–12 IPV6 Transport Layer Metrics

Metric	Description
IPV6: Datagrams Per Sec	Overall transmission rate for IP datagrams being sent and received over the network interfaces. This is the sum of Datagrams Sent Per Sec and Datagrams Received Per Sec.
IPV6: Datagrams Received Per Sec	Rate at which IP datagrams are received from the network interfaces. This counter does not include datagrams forwarded to another server.
IPV6: Datagrams Sent Per Sec	Rate at which IP datagrams are sent to the network interfaces. This counter does not include datagrams forwarded to another server.

10.13 Logical Disk Metrics

The metrics in the Logical Disk category provide information about the logical drives in the system, such as C:\.

Default Collection Interval — Every 15 minutes

Table 10–13 Logical Disk Metrics

Metric	Description
Logical Disk: Average Disk Bytes Per Transfer	Speed of the disk drives.
Logical Disk: Average Disk Queue Length	Performance of the disk. The number of disk commands waiting in the queue is normally the factor that slows disk performance by increasing the average disk queue time.
Logical Disk: Percent Disk Time	Amount of processor time spent serving disk requests. Measured against Processor:% processor time, it indicates whether disk requests are consuming processor time. If over 90%, the disk or controller is a bottleneck.

10.14 Memory Metrics

The metrics in the Memory category represent calculated metrics that described the behavior of physical and virtual memory on the computer.

Default Collection Interval — Every 15 minutes

Table 10–14 Memory Metrics

Metric	Description and User Action
Available Mega Bytes	Total physical memory available to the operating system. This amount of available memory is compared with the memory required to run all of the processes and applications on your server. Try to keep at least 10 percent of memory available for peak use. Keep in mind that by default, IIS 5.0 uses up to 50 percent of available memory for its file cache, leaving the remaining memory available for other applications running on the server.
Cache Bytes	Current size in bytes of the file system cache. By default, the cache uses up to 50 percent of available physical memory. The counter value is the sum of Memory\System Cache Resident Bytes, Memory\System Driver Resident Bytes, Memory\System Code Resident Bytes, and Memory\Pool Paged Resident Bytes.
Committed Bytes	Amount of committed virtual memory in bytes. Committed memory is the physical memory which has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.

Table 10–14 (Cont.) Memory Metrics

Metric	Description and User Action
Page Faults Per Sec	Memory bottleneck due to page faults. If a process requests a page in memory and the system cannot find it at the requested location, this constitutes a page fault. If the page is elsewhere in memory, it is called a soft page fault. If the page must be retrieved from disk, it is called a hard page fault. Most processors can handle large numbers of soft page faults without consequences, but hard page faults can cause significant delays. If the number of hard page faults is high, you might have dedicated too much memory to the caches, not leaving enough memory for the rest of the system. Sustained hard page fault rates of over five per second are a key indicator of insufficient RAM. Try increasing the amount of RAM on your server or lowering cache sizes. Other counters that can indicate a memory bottleneck are Memory:Pages Input/sec, Memory:Page Reads/sec, and Memory:Pages per second.
Page Reads Per Sec	Number of times the disk was read to resolve hard page faults.
Pages Input Per Sec	Total number of pages read from disk to resolve hard page faults.
Pages Output Per Sec	Rate at which pages are written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory, so they are likely to hold data, not code. A high rate of pages output might indicate a memory shortage. Windows writes more pages back to disk to free up space when physical memory is in short supply. This counter shows the number of pages, and can be compared to other counts of pages without conversion.
Pages Per Sec	Number of pages retrieved per second. The number should be less than one per second.
Pool Nonpaged Bytes	Monitor the pool space for all processes on the server.
Pool Paged Bytes	Shows the size, in bytes, of the paged pool. Pool Paged Bytes is calculated differently than Process\Pool Paged Bytes, so it might not equal Process(_Total)\Pool Paged Bytes.
Pool Paged Resident Bytes	Current size in bytes of the paged pool. The paged pool is an area of system memory (physical memory used by the operating system) reserved for objects that can be written to disk when they are not in use. The space used by the paged and nonpaged pools is taken from physical memory; thus, a pool that is too large denies memory space to processes.
System Cache Resident Bytes	Current size in bytes of the pageable operating system code in the file system cache. This value includes only current physical pages and excludes any virtual memory pages not currently resident.
System Code Resident Bytes	Current size in bytes of the operating system code currently in physical memory than can be written to disk when not in use. This value is a component of Memory\System Code Total Bytes.
System Driver Resident Bytes	The current size in bytes of the pageable physical memory in use by device drivers. This represents the working set (physical memory area) of the drivers. This value is a component of Memory\System Driver Total Bytes.
Transition Faults Per Sec	Rate at which page faults are resolved by recovering pages without incurring additional disk activity. Transition faults, which measure soft page faults, are counted in numbers of faults because only one page is faulted in each operation; the number of transition faults is equal to the number of pages faulted.

10.15 NBT Connection Metrics

The metrics in this category provide information about the network connectivity. If NBT connection: Bytes total/sec is close to the bandwidth of your network adapter and the other two performance counters are moderate, the network connection may be a bottleneck.

Default Collection Interval — Every 15 minutes

Table 10–15 NBT Connection Metrics

Metric	Description
Bytes Total Per Sec	Rate at which bytes are sent and received over each network adapter, including framing characters. Network Interface\\Bytes Received/sec is a sum of Network Interface\\Bytes Received/sec and Network Interface\\Bytes Sent/sec.

10.16 Network Interface Metrics

The TCP performance object consists of counters that measure the rates at which Transmission Control Protocol (TCP) segments are sent and received using TCP. It includes metrics that monitor the number of TCP connections in each TCP connection state.

Default Collection Interval — Every 15 minutes

Table 10–16 Network Interface Metrics

Metric	Description
Bytes Received Per Sec	Rate in seconds at which bytes are sent over this network adapter. Counted bytes include framing characters. This counter is a subset of Network Interface\Bytes Total/sec.
Bytes Sent Per Sec	Rate at which bytes are sent over each network adapter, including framing characters. Network Interface\\Bytes Sent/sec is a subset of Network Interface\\Bytes Total/sec.
Bytes Total Per Sec	Rate at which bytes are sent and received over each network adapter, including framing characters. Network Interface\\Bytes Received/sec is a sum of Network Interface\\Bytes Received/sec and Network Interface\\Bytes Sent/sec.
Packets Received Per Sec	Rate at which packets are received on the network interface.
Packets Sent Per Sec	Rate at which packets are sent on the network interface.

10.17 NNTP Service Metrics

The metrics in this category monitor posting, authentication, and connection activity on a Network News Transport Protocol (NNTP) Server. The NNTP service is an optional component of Internet Information Services (IIS).

Default Collection Interval — Every 15 minutes

Table 10–17 NNTP Service Metrics

Metric	Description
Articles Deleted Per Sec	Rate in incidents per second at which articles were deleted from the NNTP Server since the NNTP service was last started.
Articles Posted Per Sec	Rate in incidents per second at which articles were posted to the NNTP Server since the NNTP service was last started.
Articles Received Per Sec	Rate in incidents per second at which files were received by the NNTP Server since the NNTP service was last started.
Articles Sent Per Sec	Rate in incidents per second at which files were sent by the NNTP Server since the NNTP service was last started.
Control Messages Failed	Total number of control messages failed or not applied by the NNTP Server since the NNTP service was last started.
Current Anonymous Users	Number of users who currently have an anonymous connection using the FTP service.
Current Connections	Current number of connections established with the FTP service.

Table 10–17 (Cont.) NNTP Service Metrics

Metric	Description
Current Non Anonymous Users	Number of users who currently have a non anonymous connection using the FTP service.
Maximum Connections	Maximum number of simultaneous connections established with the FTP service.
Moderated Postings Failed	Total number of moderated postings the NNTP Server failed to send to a Simple Mail Transfer Protocol (SMTP) Server since the NNTP service was last started.
Total Passive Feeds	Number of passive feeds accepted by the NNTP Server since the NNTP service was last started.
Total Pull Feeds	Number of pull feeds made by the NNTP Server since the NNTP service was last started.
Total Push Feeds	Number of push feeds made by the NNTP Server since the NNTP service was last started.

10.18 Paging File Metrics

The metrics in this category are useful in examining paging file usage. Paging files store pages of memory used by a process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent processes from allocating memory.

Default Collection Interval — Every 15 minutes

Table 10–18 Paging File Metrics

Metric	Description
Percent Usage	Percentage of the page file instance that is in use.
Percent Usage Peak	Peak usage of the page file instance expressed as a percentage of total file size.

10.19 Physical Disk Metrics

The PhysicalDisk performance object consists of counters that monitor hard or fixed disk drives. Disks store file, program, and paging data. They are read to retrieve these items, and are written to record changes to them. The values of physical disk counters are sums of the values of the logical disks (or partitions) into which they are divided.

Default Collection Interval — Every 15 minutes

Table 10–19 Physical Disk Metrics

Metric	Description
Physical Disk: Average Disk Bytes Per Transfer	Average number of bytes that were transferred to or from the disk during write or read operations.
Physical Disk: Average Disk Queue Length	Average number of both read and write requests queued for the selected disk during the sample interval.
Physical Disk: Percent Disk Time	Percentage of elapsed time the selected disk drive was busy servicing read or write requests.

10.20 Process Metrics

The metrics in this category monitor running application program and system processes. All the threads in a process share the same address space and have access to the same data.

Default Collection Interval — Every 15 minutes

Table 10–20 Process Metrics

Metric	Description and User Action
Handle Count	Total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Page File Bytes	Current amount of virtual memory in bytes that a process has reserved for use in the paging file(s). Paging files store pages of memory used by the process. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory. If there is no paging file, this metric reflects the current amount of virtual memory that the process has reserved for use in physical memory.
Page File Bytes Peak	Maximum amount of virtual memory in bytes that a process has reserved for use in the paging file(s). Paging files store pages of memory used by the process. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory. If there is no paging file, this counter reflects the maximum amount of virtual memory that the process has reserved for use in physical memory.
Percent Processor Time	Percentage of time that the processor was executing a non-idle thread. It is calculated by measuring the duration that the idle thread is active during the sample interval, and subtracting that time from 100%. (Each processor has an idle thread that consumes cycles when no other threads are ready to run.) This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. Code executed to handle some hardware interrupts and trap conditions are included in this count. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Private Bytes	Size in bytes that this process has allocated that cannot be shared with other processes. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Thread Count	Number of threads that were active in this process. A thread is the object that executes instructions, which are the basic units of execution in a processor. Every running process has at least one thread. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Virtual Bytes	Size in bytes of the virtual address space that the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and by using too much space, the process can limit its ability to load libraries. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Working Set	Current size of the memory area that the process is utilizing for code, threads, and data. The size of the working set grows and shrinks as the VMM permits. When memory is becoming scarce, the working sets of the applications are trimmed. When memory is plentiful, the working sets are allowed to grow. Larger working sets mean more code and data in memory, which increases the overall performance of the applications. However, a large working set that does not shrink appropriately is usually an indication of a memory leak.

10.21 Processor Metrics

The metrics in this category measure aspects of processor activity. The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The Processor object represents each processor as an instance of the object.

Default Collection Interval — Every 15 minutes

Table 10–21 Processor Metrics

Metric	Description
Interrupts Per Sec	Rate in incidents per second at which the processor received and serviced hardware interrupts.
Percent DPC Time	Percentage of time the processor received and serviced deferred procedure calls (DPCs) during the sample interval. DPCs are interrupts that run at a lower priority than standard interrupts.
Percent Processor Time	Percentage of elapsed time that this thread used the processor to execute instructions.

10.22 SMTP Service Metrics

The metrics in this category describe the activity of the Exchange NTFS store driver, which is responsible for storing queued messages that the Simple Mail Transfer Protocol (SMTP) service is processing.

Default Collection Interval — Every 15 minutes

Table 10–22 SMTP Service Metrics

Metric	Description
Average Retries Per Message Delivered	Average number of retries per local delivery.
Average Retries Per Message Sent	Number of retries per outbound message sent.
Bad Mailed Messages General Failure	Number of messages sent to badmail for reasons not associated with a specific counter.
Bytes Received Per Sec	Rate at which the SMTP server is receiving data in bytes per second.
Bytes Sent Per Sec	Rate at which the SMTP server is sending data in bytes per second.
Bytes Total Per Sec	Rate at which the SMTP server is sending and receiving data in bytes per second (sum of SMTP Service\Bytes Sent/sec and SMTP Service\Bytes Received/sec).
Cat LDAP Connections	Total number of LDAP connections opened since the computer was last started.
Cat LDAP Search Failures	Number of failures to dispatch an asynchronous LDAP search.
Cat LDAP Searches Per Sec	Number of LDAP searches that were successfully dispatched.
Cat Messages Submitted Per Sec	Rate in incidents per second at which messages were submitted to the organizer.
Categorizer Queue Length	Number of messages in the Simple Mail Transfer Protocol (SMTP) Categorizer queue waiting for directory service attribute searches using global catalog servers. As a general guideline, the maximum value should be less than 10.
Connection Errors Per Sec	Rate, in incidents per second, at which connection errors occurred.
DNS Queries Per Sec	Rate in incidents per second of DNS lookups.

Table 10–22 (Cont.) SMTP Service Metrics

Metric	Description
Local Queue Length	Number of messages in the local queue.
Local Retry Queue Length	Number of messages in the local retry queue.
Message Delivery Retries	Total number of local deliveries that were retried.
Message Send Retries	Total number of outbound message sends that were retried.
Messages Delivered Per Sec	Rate in incidents per second at which messages were delivered to local mailboxes.
Messages Received Per Sec	Rate, in incidents per second, at which inbound messages were being received.
Messages Sent Per Sec	Rate in incidents per second at which outbound messages were sent.
Remote Queue Length	Number of messages that were in the remote queue.
Remote Retry Queue Length	Number of messages that were in the retry queue for remote delivery.
Routing Table Lookups Per Sec	Rate in incidents per second of routing table lookups.
Total Connection Errors	Total number of connection errors.
Total DSN Failures	Total number of failed DSN generation attempts.

10.23 System Metrics

The System performance object consists of counters that apply to more than one component of the computer. The data collected by the system counters is derived from activity in the processor, memory, or disk subsystems.

Default Collection Interval — Every 15 minutes

Table 10–23 System Metrics

Metric	Description and User Action
Context Switches Per Sec	Shows the combined rate in incidents per second at which all processors on the computer were switched from one thread to another. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements.
Processor Queue Length	Number of threads in the processor queue. Unlike the disk counters, this counter shows ready threads only, not threads that are running.
System Calls Per Sec	Combined rate in incidents per second of calls to operating system service routines by all processes running on the computer.

10.24 TCPV4 Network Layer Metrics

The TCP performance object consists of counters that measure the rates at which Transmission Control Protocol (TCP) segments are sent and received using TCP. It includes counters that monitor the number of TCP connections in each TCP connection state.

Default Collection Interval — Every 15 minutes

Table 10–24 TCPV4 Network Layer Metrics

Metric	Description
TCPV4: Connection Established	Number of TCP connections for which the state was either ESTABLISHED or CLOSE-WAIT since the server was last started.
TCPV4: Connection Failures	Number of times that TCP connections have directly transitioned to the CLOSED state from the SYN-SENT or SYN-RCVD state, plus the number of times TCP connections have directly transitioned to the LISTEN state from the SYN-RCVD state since the server was last started.
TCPV4: Connection Reset	Number of times that TCP connections have directly transitioned to the CLOSED state from either the ESTABLISHED or CLOSE-WAIT state since the server was last started.
TCPV4: Segments Per Sec	Rate in incidents per second at which TCP segments were sent or received using the TCP protocol. Segments/sec is the sum of the values of Segments Received/sec and Segments Sent/sec.
TCPV4: Segments Received Per Sec	Rate in incidents per second at which segments were received, including those received in error. This count includes segments received on currently established connections. Segments Received/sec is a subset of Segments/sec.
TCPV4: Segments Retransmitted Per Sec	Rate in incidents per second at which segments containing one or more previously transmitted bytes were retransmitted.
TCPV4: Segments Sent Per Sec	Rate in incidents per second at which segments were sent. This value includes those on current connections, but excludes those containing only retransmitted bytes. Segments Sent/sec is a subset of Segments/sec.

10.25 TCPV6 Network Layer Metrics

The TCP performance object consists of counters that measure the rates at which Transmission Control Protocol (TCP) segments are sent and received using TCP. It includes counters that monitor the number of TCP connections in each TCP connection state.

Default Collection Interval — Every 15 minutes

Table 10–25 TCPV6 Network Layer Metrics

Metric	Description
TCPV6: Connection Established	Number of TCP connections for which the state was either ESTABLISHED or CLOSE-WAIT since the server was last started.
TCPV6: Connection Failures	Number of times that TCP connections have directly transitioned to the CLOSED state from the SYN-SENT or SYN-RCVD state, plus the number of times TCP connections have directly transitioned to the LISTEN state from the SYN-RCVD state since the server was last started.
TCPV6: Connection Reset	Number of times that TCP connections have directly transitioned to the CLOSED state from either the ESTABLISHED or CLOSE-WAIT state since the server was last started.
TCPV6: Segments Per Sec	Rate in incidents per second at which TCP segments were sent or received using the TCP protocol. Segments/sec is the sum of the values of Segments Received/sec and Segments Sent/sec.
TCPV6: Segments Received Per Sec	Rate in incidents per second at which segments were received, including those received in error. This count includes segments received on currently established connections. Segments Received/sec is a subset of Segments/sec.
TCPV6: Segments Retransmitted Per Sec	Rate in incidents per second at which segments containing one or more previously transmitted bytes were retransmitted.
TCPV6: Segments Sent Per Sec	Rate in incidents per second at which segments were sent. This value includes those on current connections, but excludes those containing only retransmitted bytes. Segments Sent/sec is a subset of Segments/sec.

10.26 Thread Performance Metrics

The Thread performance object consists of counters that measure aspects of thread behavior. A thread is the basic object that executes instructions on a processor. All running processes have at least one thread.

Default Collection Interval — Every 15 minutes

Table 10–26 Thread Performance Metrics

Metric	Description
Context Switches Per Sec	Combined rate in incidents per second at which all processors on the computer were switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, or is preempted by a higher priority ready thread.
Percent Processor Time	Percentage of time all threads are using the processors.

10.27 WWW Service Metrics

The Web Service performance object counters (installed with Internet Information Services) monitor file transfer rates, bandwidth usage, connection rates, errors, and numbers and types of users. You can view performance data for all instances of this object (using the _Total instance) or for specific instances, such as the Default Web Site or Administration Web Site instances.

Default Collection Interval — Every 15 minutes

Table 10–27 WWW Service Metrics

Metric	Description and User Action
400 Series Errors	Total errors generated in the 400 series, such as 404, 403, and so forth.
404 Errors	Number of requests since the service started that the server did not satisfy because the requested document was not found. This type of request is usually reported to the client as an HTTP 404 error message.
423 Errors	Number of requests since the service started that the server did not satisfy because the requested document was not locked. This type of request is usually reported to the client as an HTTP 423 error message.
Bytes Received Per Sec	Rate in incidents per second at which data bytes were received by the web service.
Bytes Sent Per Sec	Rate in incidents per second at which data bytes were sent by the web service.
Bytes Total Per Sec	Sum of Web Service\Bytes Sent/sec and Web Service\Bytes Received/sec. This is the total rate in incidents per second at which bytes were transferred by the Web service.
CGI Requests	Rate in incidents per second at which the Common Gateway Interface (CGI) requests were being simultaneously processed by the Web service.
Current Anonymous Users	Number of users who had an anonymous connection using the Web service.
Current CGI Requests	Current number of CGI requests that were being simultaneously processed by the Web service.
Current Connections	Current number of connections established with the Web service. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements for the adequate current connections number.
Current ISAPI Extension Requests	Current number of Internet Server Application Programming Interface (ISAPI) extension requests that were being simultaneously processed by the Web service.
Current Non Anonymous Users	Number of users who currently have a non-anonymous connection using the Web service.

Table 10–27 (Cont.) WWW Service Metrics

Metric	Description and User Action
Files Per Sec	Rate in incidents per second at which files were transferred; that is, sent and received by the Web service.
Files Received Per Sec	Rate in incidents per second at which files were received by the Web service.
Files Sent Per Sec	Rate in incidents per second at which files were sent by the Web service.
Get Requests	Number of incidents whereby HTTP requests were made on the server using the GET method.
Get Requests Per Sec	Rate in incidents per second at which HTTP requests were made using the GET method. GET requests are generally used for basic file retrievals or image maps, though they can be used with forms.
Head Requests	Rate HTTP requests are made using the HEAD method. HEAD requests generally indicate that a client is querying the state of a document it already has so that the client can determine if the document needs to be refreshed.
Head Requests Per Sec	Rate in incidents per second at which HTTP requests are made using the HEAD method. HEAD requests generally indicate that a client is querying the state of a document it already has so that the client can determine if the document needs to be refreshed.
ISAPI Extension Requests Per Sec	Rate in incidents per second at which ISAPI Extension Requests were being simultaneously processed by the Web service.
Maximum Connections	Maximum number of simultaneous connections established with the Web service.
Post Requests	Number of HTTP requests using the POST method, which is generally used for forms or gateway requests.
Post Requests Per Sec	Rate in incidents per second at which HTTP requests using the POST method were made. POST requests are generally used for forms or gateway requests.
Total CGI Requests	Total number of CGI requests.
Total Connection Attempts (All Instances)	Number of connections attempted using the Web service (counted since service startup). This counter is for all instances listed.
Total Get Requests	Total number of HTTP requests using the GET method (counted since service startup).
Total Head Requests	Total number of HTTP requests using the HEAD method (counted since service startup). HEAD requests generally indicate that a client is querying the state of a document it already has to determine if the document needs to be refreshed.
Total ISAPI Extension Requests	Total number of ISAPI Extension Requests. ISAPI Extension Requests are custom gateway dynamic-link libraries (DLLs) that the administrator can install to add forms processing or other dynamic data sources. The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements for the adequate Total ISAPI Extension Requests.
Total Locked Errors	Total number of requests that could not be satisfied by the server because the requested document was locked. These are generally reported as an HTTP 423 error code to the client. The count is the total since service startup.
Total Method Requests	Total number of all HTTP requests (counted since service startup). The default warning and critical threshold values for this metric are set to an 'UnDefined' value. You can provide a value for the warning and critical thresholds based on your current environment and your requirements for the adequate "total method requests."

Table 10–27 (Cont.) WWW Service Metrics

Metric	Description and User Action
Total Method Requests Per Sec	Rate in incidents per second at which all HTTP requests were made.
Total Not Found Errors	Total number of requests that could not be satisfied by the server because the requested document could not be found. These are generally reported as an HTTP 404 error code to the client. The count is the total since service startup.
Total Post Requests	Total number of HTTP requests using the POST method (counted since service startup). POST requests are generally used for forms or gateway requests.

10.28 WWW Service Cache Metrics

The WWW service and FTP service do not share a common cache. Instead, the caches are split into two separate performance objects: one for FTP service and one for the WWW service. WWW service cache counters only monitor server performance. Therefore, you cannot configure them to monitor individual sites.

Table 10–28 WWW Service Cache Metrics

Metric	Description
Active Flushed Entries	Number of flushed user-mode cache entries, though memory is still allocated for these entries. The allocated memory is released after all current transfers are complete.
File Cache Hits Percent	Ratio of user-mode file cache hits to total cache requests that have been made since the WWW service started up.
Kernel: URI Cache Flushes	Total number of URI cache flushes that have occurred since the WWW service started.
Kernel: URI Cache Hits Percent	Ratio of URI cache hits to total cache requests that have occurred since the WWW service started.
Kernel: URI Cache Misses	Number of unsuccessful lookups that have been made in the user-mode URI cache since the WWW service started.

10.29 WWW Service Worker Process and ASP Error Events Metrics

The metrics in this category provide information about the errors pertaining to the worker process and ASP errors.

Default Collection Interval — Every 15 minutes

Table 10–29 WWW Service Worker Process and ASP Error Events Metrics

Metric	Description
Description	Description message for the error.
Windows Event Severity	Severity of the error.

10.30 WWW Service Worker Process and ASP Warning Events Metrics

The metrics in this category provide information about the warnings pertaining to the worker process and ASP errors.

Default Collection Interval — Every 15 minutes

Table 10–30 WWW Service Worker Process and ASP Warning Events Metrics

Metric	Description
Description	Description message for the warning.
Windows Event Severity	Severity of the warning.

11

Microsoft Internet Security and Acceleration Metrics

This chapter provides descriptions for all Microsoft Internet Security and Acceleration (ISA) metric categories, and tables list and describe associated metrics for each category. The tables also provide user actions if any of the metrics for a particular category support user actions.

11.1 Firewall Packet Engine Metrics

The metrics in this category provide performance information for the firewall engine.

Table 11–1 Firewall Packet Engine Metrics

Metric	Description and User Action
Active Connections	Total number of active connections currently passing data.
Allowed Packets Per Sec	Number of packets per second that the ISA server allows to pass.
Bytes Per Sec	Number of bytes passed through the ISA server per second.
Connections Per Sec	Number of new connections created per second.
Dropped Packets Per Sec	Number of packets the ISA server dropped per second. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Packets Per Sec	Number of packets the ISA server inspected per second. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
TCP Established Connections Per Sec	Number of Transmission Control Protocol (TCP) connections newly established per second. A TCP connection is counted as established after the 3-way SYN handshake was completed successfully.

11.2 Firewall Service Metrics

The metrics in the this category provide performance information about the firewall service.

Table 11–2 Firewall Service Metrics

Metric	Description and User Action
Accepting TCP Connections	Number of connection objects that wait for a Transmission Control Protocol (TCP) connection from firewall clients.
Active Sessions	Number of active sessions for the firewall service.
Active TCP Connections	Total number of active TCP connections currently passing data. Connections pending or not yet established are counted elsewhere.
Active UDP Connections	Total number of active User Datagram Protocol (UDP) connections.
Available UDP Mappings	Number of available UDP mappings.
Available Worker Threads	Number of firewall worker threads that are available or waiting in the completion port queue. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Bytes Read Per Sec	Number of bytes read by the data pump per second.
Bytes Written Per Sec	Number of bytes written by the data pump per second.
DNS Cache Entries	Current number of DNS domain name entries cached as result of firewall service activity. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
DNS Cache Flushes	Number of times the firewall service has flushed or cleared the DNS domain name cache. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
DNS Cache Hits	Total number of times the firewall service found a DNS domain name within the DNS cache. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
DNS Cache Hits %	Percentage of DNS domain names serviced by the DNS cache from the total of all DNS entries that the firewall service has retrieved. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
DNS Retrievals	Total number of DNS domain names that the firewall service has retrieved. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Failed DNS Resolutions	Number of `gethostbyname` and `gethostbyaddr` API calls that have failed. These calls resolved host DNS domain names and IP addresses for firewall service connections. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Kernel Mode Data Pumps	Number of kernel mode data pumps the firewall service created.
Listening TCP Connections	Number of connection objects that waited for TCP connections from remote Internet computers.

Table 11–2 (Cont.) Firewall Service Metrics

Metric	Description and User Action
Pending DNS Resolutions	Number of pending DNS resolutions.
Pending TCP Connections	Number of pending top connections.
Secure NAT Mappings	Number of mappings created by SecureNAT.
Successful DNS Resolutions	Number of successful DNS name resolutions.
TCP Bytes Transferred Per Sec By Kernel Mode Data Pump	Number of bytes transferred by TCP per second via the kernel mode data pump.
TCP Connections Awaiting Inbound Connect Call To Finish	Number of TCP connections awaiting an inbound connection call to finish. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
UDP Bytes Transferred Per Sec By Kernel Mode Data Pump	Number of bytes transferred by TCP per second via the kernel mode data pump.
Worker Threads	Number of currently active firewall worker threads.

11.3 H.323 Filter Metrics

The metrics in this category provide information about the active and total calls to the H.323 filter.

Table 11–3 H.323 Filter Metrics

Metric	Description and User Action
Active H.323 Calls	Displays currently active H.323 calls. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Total H.323 Calls	Displays all H.323 calls handled by the H.323 filter since the ISA server computer was started.

11.4 ISA Server Error Events Metrics

The metrics in this category provide information about the error events the ISA server generates.

Table 11–4 ISA Server Error Events Metrics

Metric	Description
Date-Time	Date and time when the error was generated.
Description	Description text of the error that is mentioned in the event log file.
Event ID	Every error generated has an event ID or record number. Using this event ID, you can search for more information on the web and MSN technet.
Log Name	Name of the log file where the error was generated.
Source	Component that generated the error (that is, NTDS intersite messaging).
Windows Event Security	Severity of the error.

11.5 ISA Server Warning Events Metrics

The metrics in this category provide information about the warning events the ISA server generates.

Table 11–5 ISA Server Warning Events Metrics

Metric	Description
Date-Time	Date and time when the warning was generated.
Description	Description text of the warning that is mentioned in the event log file.
Event ID	Every warning generated has an event ID or record number. Using this event ID, you can search for more information on the web and MSN technet.
Log Name	Name of the log file where the warning was generated.
Source	Component that generated the warning (that is, NTDS intersite messaging).
Windows Event Severity	Severity of the warning.

11.6 ISASTGCTRL Server Error Events Metrics

The metrics in this category show the error events generated by the ISASTGCTRL service in the event log file. The Windows ISASTGCTRL service manages read and write access to the Configuration Storage server information.

Table 11–6 ISASTGCTRL Server Error Events Metrics

Metric	Description
Date-Time	Date and time when the error was generated.
Description	Description text of the error that is mentioned in the event log file.
Event ID	Every error generated has an event ID or record number. Using this event ID, you can search for more information on the web and MSN technet.
Log Name	Name of the log file where the error was generated.
Source	Component that generated the error (that is, NTDS intersite messaging).
Windows Event Severity	Severity of the error.

11.7 ISASTGCTRL Server Warning Events Metrics

The metrics in this category show the warning events generated by the ISASTGCTRL service in the event log file. The Windows ISASTGCTRL service manages read and write access to the Configuration Storage server information.

Table 11–7 ISA Server Warning Events Metrics

Metric	Description
Date-Time	Date and time when the warning was generated.
Description	Description text of the warning that is mentioned in the event log file.
Event ID	Every warning generated has an event ID or record number. Using this event ID, you can search for more information on the web and MSN technet.
Log Name	Name of the log file where the warning was generated.
Source	Component that generated the warning (that is, NTDS intersite messaging).
Windows Event Severity	Severity of the warning.

11.8 Process Metrics

The metrics in this category provide information about the important ISA server processes.

Table 11–8 Process Metrics

Metric	Description and User Action
Creating Process ID	Process ID of the process that created the process.
Elapsed Time	Total elapsed time, in seconds, that this process has been running.
Handle Count	Total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in the process. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
ID Process	Unique identifier of this process. ID process numbers are reused, so they only identify a process for the lifetime of that process.
IO Data Bytes Per Sec	Rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Data Operations Per Sec	Rate at which the process is issuing read and write I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Other Bytes Per Sec	Rate at which the process is issuing bytes to I/O operations that do not involve data such as control operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Other Operations Per Sec	Rate at which the process is issuing I/O operations that are neither read nor write operations (for example, a control function). This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Read Bytes Per Sec	Rate at which the process is reading bytes from I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Read Operations Per Sec	Rate at which the process is issuing read I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Write Bytes Per Sec	Rate at which the process is writing bytes to I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
IO Write Operations Per Sec	Rate at which the process is issuing write I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.
Page Faults Per Sec	Rate at which page faults are occurring from the threads executing in this process. A page fault occurs when a thread refers to a virtual memory page that is not in its working set in main memory. This may not cause the page to be fetched from disk if it is on the standby list and therefore already in main memory, or if it is in use by another process with which the page is shared.
Page File Bytes	Current amount of virtual memory, in bytes, that this process has reserved for use in the paging file(s). Paging files store pages of memory used by the process that are not contained in other files. All processes share paging files, and the lack of space in paging files can prevent other processes from allocating memory. If there is no paging file, this counter reflects the current amount of virtual memory that the process has reserved for use in physical memory.
Page File Bytes Peak	Maximum amount of virtual memory, in bytes, that this process has reserved for use in the paging file(s). Paging files store pages of memory used by the process that are not contained in other files. Paging files are shared by all processes, and the lack of space in paging files can prevent other processes from allocating memory. If there is no paging file, this counter reflects the maximum amount of virtual memory that the process has reserved for use in physical memory.

Table 11–8 (Cont.) Process Metrics

Metric	Description and User Action
Percent Privileged Time	Percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service often runs in privileged mode to gain access to system-private data. This data is protected from access by threads executing in user mode. Calls to the system can be explicit or implicit, such as page faults or interrupts. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Percent Processor Time	Percentage of elapsed time that all of process threads used the processor to execute instructions. An instruction is the basic unit of execution in a computer, a thread is the object that executes instructions, and a process is the object created when a program is run. Code executed to handle some hardware interrupts and trap conditions are included in this count. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Percent User Time	Percentage of elapsed time that the process threads spent executing code in user mode. Applications, environment subsystems, and integral subsystems execute in user mode. Code executing in user mode cannot damage the integrity of the Windows executive, kernel, and device drivers. Unlike some early operating systems, Windows uses process boundaries for subsystem protection in addition to the traditional protection of user and privileged modes. Some work done by Windows on behalf of the application might appear in other subsystem processes in addition to the privileged time in the process.
Private Bytes	Current size, in bytes, of memory that this process has allocated that cannot be shared with other processes. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Thread Count	Number of threads currently active in this process. An instruction is the basic unit of execution in a processor, and a thread is the object that executes instructions. Every running process has at least one thread.
Virtual Bytes	Current size, in bytes, of the virtual address space the process is using. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. Virtual space is finite, and the process can limit its ability to load libraries. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Virtual Bytes Peak	Maximum size, in bytes, of virtual address space the process has used at any one time. Use of virtual address space does not necessarily imply corresponding use of either disk or main memory pages. However, virtual space is finite, and the process might limit its ability to load libraries.
Working Set	Current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they will then be soft-faulted back into the Working Set before leaving main memory.
Working Set Peak	Maximum size, in bytes, of the Working Set of this process at any point in time. The Working Set is the set of memory pages touched recently by the threads in the process. If free memory in the computer is above a threshold, pages are left in the Working Set of a process even if they are not in use. When free memory falls below a threshold, pages are trimmed from Working Sets. If they are needed, they will then be soft-faulted back into the Working Set before they leave main memory.

11.9 Web Proxy Service Metrics

The Web Proxy Service enables any Worldwide Web client to access internet resources using the HTTP, HTTPS, Gopher, and FTP protocols on behalf of the client.

Table 11–9 Web Proxy Service Metrics

Metric	Description and User Action
Array Bytes Received Per Sec	Tracks the rate at which data bytes are received from other ISA Server computers within the same array.
Array Bytes Sent Per Sec	Tracks the rate at which data bytes are sent from other ISA Server computers within the same array.
Average Current Array Fetches Time Per Request (Millisec)	Represents the sum of Array Bytes Sent/Sec and Array Bytes Received/Sec. This is the total rate for all data bytes transferred between the ISA Server computer and other members of the same array.
Cache Hit Percent Ratio	Determines how many Web Proxy client requests have been served using cached data (Total Cache Fetches) as a percentage of the total number of successful Web Proxy client requests to the ISA Server computer (Total Successful Requests). Its value provides a good indication of cache effectiveness. A high metric indicates that a high level of requests are being serviced from the cache, meaning faster response times. A zero metric indicates that caching is not enabled. A low metric may indicate a configuration problem. The cache size may be too small, or requests may not be cacheable. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Cache Running Hit Ratio	Measures the amount of requests served from the cache as a percentage of total successful requests serviced. This ratio is the same as that measured by Cache Hit Ratio(%). The difference between these two metrics is that Cache Running Hit Ratio measures this ratio for the last 10,000 requests serviced, and Cache Hit Ratio measures this ratio since the last time that the Web Proxy service started. This means that Cache Running Hit Ratio provides a more dynamic evaluation of cache effectiveness.
Client Bytes Received Per Sec	Rate at which data bytes are received from Web Proxy clients. The value changes according to the volume of Web Proxy client requests, but a consistently slow rate may indicate a delay in servicing requests.
Client Bytes Sent Per Sec	Rate at which data bytes are sent to Web Proxy clients. The value changes according to the volume of Web Proxy client requests, but a consistently slow rate may indicate a delay in servicing requests.
Current Array Fetches Average Time Per Request (Millisec)	Provides the mean number of milliseconds required to service a Web Proxy client request that is fetched through another array member. This does not include requests for services by the Secure Sockets Layer (SSL) tunnel.
Current Average Time Per Request (Millisec)	Represents the mean number of milliseconds required to service a Web Proxy client request, not including requests serviced by the SSL tunnel. This counter can be monitored at peak and off-peak times for a comprehensive picture of how fast client requests are being serviced. A counter that is too high might indicate that the ISA Server is having difficulty handling all requests and that requests are being delayed.
Current Cache Fetches Average Time Per Request (Millisec)	Mean number of milliseconds required to service a Web Proxy client request from cache. This does not include requests for services by the Secure Sockets Layer (SSL) tunnel. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Current Direct Fetches Average Time Per Request (Millisec)	Mean number of milliseconds required to service a Web Proxy client request directly to the Web server or upstream proxy. This does not include requests for services by the Secure Sockets Layer (SSL) tunnel. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.

Table 11–9 (Cont.) Web Proxy Service Metrics

Metric	Description and User Action
DNS Cache Entries	Details the current number of DNS domain name entries cached by the Web Proxy service. A high metric suggests a beneficial impact on performance, since a DNS cache entry eliminates the need for a DNS lookup, saving system resources.
DNS Cache Flushes	Details the total number of times that the DNS domain name cache has been flushed or cleared by the Web Proxy service. When there is no room left for more data in the DNS cache, the DNS cache is flushed to allow new entries to be made.
DNS Cache Hits	Tracks the total number of times the Web Proxy service found a DNS domain name within the DNS cache. This metric can be compared with previous DNS counters to find out if DNS caching is working efficiently. A low number of DNS cache hits impact performance, as every DNS lookup slows performance, particularly if a problem arises in the lookup process.
DNS Cache Hits Percent	Determines how many DNS entries have been resolved using cached data (DNS cache hits) as a percentage of the total number of DNS domain names retrieved by the Web Proxy service (DNS retrievals). A high metric means better performance because the DNS data is served from the cache, rather than incurring the overhead of resolving DNS lookups. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
DNS Retrievals	Represents the total number of DNS domain names that the Web Proxy service has retrieved.
Failing Requests Per Sec	Monitors the rate per second that Web Proxy client requests have completed with some type of error. This counter can be compared with Requests/Sec to indicate how well the ISA Server is servicing incoming Web requests. A high failure rate, in comparison to the rate of incoming requests, suggests that the ISA Server is having difficulty coping with all incoming requests. Connection settings for incoming Web requests may be incorrectly configured, or connection bandwidth may be insufficient.
FTP Requests	Tracks the number of File Transfer Protocol (FTP) requests made to the Web Proxy service. A consistently low counter may influence the caching policy for FTP objects.
Gopher Requests	Tracks the number of Gopher requests that have been made to the Web Proxy service.
HTTP Requests	Tracks the number of Hypertext Transfer Protocol (HTTP) requests that have been made to the Web Proxy service.
HTTPS Sessions	Represents the total number of Secure Hypertext Transfer Protocol (HTTPS) secured sessions serviced by the SSL tunnel.
Maximum Users	Tracks the maximum number of users that have connected to the Web Proxy service simultaneously. This counter can be useful for determining load usage and license requirements.
Requests Per Sec	Monitors the rate or incoming requests made to the Web Proxy service. A higher value means that more ISA Server resources will be required to service incoming requests.
Reverse Bytes Received Per Sec	Monitors the rate at which data bytes are received by the Web Proxy service from Web publishing servers in response to incoming requests. This rate can be monitored at peak and off-peak times as an indication of how the ISA Server is performing in servicing incoming Web requests.
Reverse Bytes Sent Per Sec	Monitors the rate at which data bytes are sent by the Web Proxy service to Web publishing servers in response to incoming requests. This rate can be monitored at peak and off-peak times as an indication of how the ISA Server is performing in servicing incoming Web requests.
Site Access Denied	Tracks the total number of Internet sites to which the Web Proxy service has denied access. An excessively high number might indicate an access policy that is too restrictive. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.

Table 11–9 (Cont.) Web Proxy Service Metrics

Metric	Description and User Action
Site Access Granted	Tracks the total number of Internet sites to which the Web Proxy service has granted access. This can be compared with Site Access Denied to provide a numeric summary of the results of access policy configuration.
SNEWS Sessions	Represents the total number of SNEWS sessions serviced by the SSL tunnel.
SSL Client Bytes Received Per Sec	Measures the rate at which SSL data bytes are received by the Web Proxy service from secured Web Proxy clients. This is similar to Client Bytes Received/Sec, but counts only SSL requests.
SSL Client Bytes Sent Per Sec	Measures the rate at which SSL data bytes are sent by the Web Proxy service to secured Web Proxy clients. This is similar to Client Bytes Sent/Sec, but counts only SSL requests.
SSL Client Bytes Total Per Sec	Represents the sum of SSL Client Bytes Sent/Sec and SSL Client Bytes Received/Sec. This is the total rate for all bytes transferred between the Web Proxy service and SSL clients.
Thread Pool Active Sessions	Represents the number of sessions that thread pool threads are actively servicing.
Thread Pool Failures	Represents the number of requests rejected because the thread pool was full. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Thread Pool Size	Represents the number of threads in the thread pool. This thread pool represents the resources available to service client requests.
Total Array Fetches	Totals the number of Web Proxy client requests served by requesting the data from another ISA Server within this array. These requests are the result of the Cache Array Routing Protocol (CARP) algorithm, which randomly stores objects in any of the member servers cache. This metric is influenced by the cache size for each ISA Server in the array, since a server with a larger cache holds more cache items. The load factor for each server can also be configured to determine how workload is divided among array members.
Total Cache Fetches	Monitors the total number of Web Proxy client requests served by using cached data. A high number indicates a cache being fully exploited.
Total Failed Requests	Represents the total number of requests that the Web Proxy service has failed to process due to errors. Errors can result from the Web Proxy service failing to locate a requested server URL on the Internet, or because the client did not have authorized access to the requested URL. This metric should be far lower than Total Successful Requests. If it is not, this indicates that the ISA Server is failing to service requests effectively. This could be a configuration problem, indicate a connection that is too slow, or indicate an access policy that is too restrictive. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Total Pending Connects	Total number of pending connections to the Web Proxy service. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Total Requests	Represents the total number of requests made to the Web Proxy service. It is the total of two other counters: Total Successful Requests and Total Failed Requests.
Total Reverse Fetches	Represents the total number of incoming requests that have been served by requesting the data from Web publishing servers.
Total SSL Sessions	Represents the total number of SSL sessions serviced by the SSL tunnel.
Total Successful Requests	Represents the total number of requests that the Web Proxy service has successfully processed. This metric can be compared with Total Requests and Total Failed Requests to indicate the effectiveness of the ISA Server in servicing requests.

Table 11–9 (Cont.) Web Proxy Service Metrics

Metric	Description and User Action
Total Upstream Fetches	Tracks the total number of requests that have been served by using data from the Internet or from a chained proxy computer. This metric can be compared to Total Cache Fetches to see what proportion of requests are being serviced from remote servers on the Internet or upstream proxies compared with those being serviced from the cache.
Total Users	Represents the total number of users that have ever connected to the Web Proxy service. It represents a history of past server usage. The default warning and critical threshold values for this metric are set to an UnDefined value. You can provide a value for the warning and critical thresholds based on your current environment and requirements.
Unknown SSL Sessions	Represents the total number of unknown SSL sessions serviced by the SSL tunnel.
Upstream Bytes Received Per Sec	Indicates the rate at which the Web Proxy service receives data bytes from remote servers on the Internet or from a chained proxy computer in response to requests from the Web Proxy service. The value of this counter partially depends on the connection bandwidth. If the metric value is consistently low, this may indicate a bottleneck caused by a slow connection. Changing the bandwidth priority configuration may help in this situation, or a faster connection may be required.
Upstream Bytes Sent Per Sec	Indicates the rate at which the Web Proxy service sends data bytes to remote servers on the Internet or to a chained proxy computer. The value of this counter partially depends on the connection bandwidth. If the metric value is consistently low, this may indicate a bottleneck caused by a slow connection. Changing the bandwidth priority configuration may help in this situation, or a faster connection may be required.
Upstream Bytes Total Per Sec	Sum of Upstream Bytes Sent/Sec and Upstream Bytes Received/Sec. It represents the total rate for all bytes transferred between the Web Proxy service and remote servers on the Internet or a chained proxy server.