Oracle® Enterprise Manager

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

Release 3 (3.0)

B28748-02

December 2006

ORACLE®

Oracle Enterprise Manager System Monitoring Plug-in Metric Reference Manual for Non-Oracle Database Management Release 3 (3.0)

B28748-02

Primary Author: Michael Zampiceni

Contents

2 IBM DB2 Database Metrics

Preface

This manual is a compilation of the plug-ins metrics provided in Oracle Enterprise Manager for database 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, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

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 SQL Server*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for IBM DB2 Database*
- *Oracle Enterprise Manager Concepts*
- *Oracle Enterprise Manager Grid Control Quick Installation Guide*
- *Oracle Enterprise Manager Grid Control Quick Installation Guide*
- *Oracle Enterprise Manager Grid Control Installation and Basic Configuration*
- *Oracle Enterprise Manager Configuration for Oracle Collaboration Suite*
- *Oracle Enterprise Manager Advanced Configuration*
- *Oracle Enterprise Manager Policy Reference Manual*
- *Oracle Enterprise Manager Extensibility*
- *Oracle Enterprise Manager Command Line Interface*
- *Oracle Enterprise Manager SNMP Support Reference Guide*
- *Oracle Enterprise Manager Licensing Information*

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

This manual contains chapters for the Microsoft SQL Server and the IBM DB2 Database. The metrics in these chapters appear in alphabetical order according to category.

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

  Suggests how to solve the problem causing the alert.

Definitions of Columns in Metric Summary Tables

The Metric Summary table in Enterprise Manager Grid Control is part of the overall metric information. The following table provides descriptions of columns in the Enterprise Manager Metric Summary table.

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.

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, when the Average File Write Time degrades to the point an alert should trigger, it could be almost 10 minutes before Enterprise Manager receives an indication of the alert. This column is present in the Metric Collection Summary table only for Oracle Database 10*g* metrics.
Collection Schedule	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 Interval	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.

After 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 is 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 appears.

1

Microsoft SQL Server Metrics

This chapter provides descriptions for all Microsoft SQL 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. Shaded rows represent key columns for a particular category.

1.1 Configuration Metrics

Configuration metrics consist of the following categories:

- SQL Server Configuration
- Registry Setting Configuration
- Database Setting Configuration

1.1.1 SQL Server Configuration Metrics

The metrics in this category represent a SQL Server installation. The metrics contain details of the product and version of the SQL Server instance.

- Table Name — MGMT_EMX_MSSQL_SQLSERVER
- View Name — MGMT_EMX_MSSQL_SQLSERVER_VIEW

Default Collection Interval — Every 24 hours

Table 1–1 SQL Server Configuration Metrics

Metric	Description
Name	Name of the SQL Server.
Clustered	Whether the server belongs to a cluster.
Package	Product installed. 0 — Unknown 1 — Office 0 — Unknown 0 — Unknown 0 — Unknown
Product	Installed product.
Version	Installed version.
Operatingsystem	Operating system on which the installation is done.

1.1.2 Registry Setting Configuration Metrics

The metrics in this category contain the installation and run-time parameters of the SQL Server stored in the registry.

- Table Name — MGMT_EMX_MSSQL_REGSETTING
- View Name — MGMT_EMX_MSSQL_REGSETTING_VIEW

Default Collection Interval — Every 24 hours

Table 1–2 Registry Setting Configuration Metrics

Metric	Description
Setting ID (key column)	Instance of the SQL Server.
Agent Log File	Path and file name for the Agent log.
Backup Directory	Location of the backup files directory.
Case Sensitive	Comparison method for multi-byte character data is either case-sensitive or not.
Error Log Path	Operating system path and file name to be used for the SQL Server error log.
Master DB Path	The full path and file name of the operating system file containing the master database.
NT Event Logging	Whether the SQL Server uses the Windows NT application log. If TRUE, the SQL Server sends all events to the Windows NT application log and the SQL Server error log. If FALSE, the SQL Server sends events only to the SQL Server error log.
Number of Processors	Number of CPUs available to the SQL Server on the server.
Perf Mon Mode	Operating system path and file name to be used for the SQL Server error log. Windows NT Performance Monitor polling behavior when the monitor is launched. 0 — Continuous 1 — On demand
Registered Organization	Company name supplied by the installer.
Registered Owner	User name supplied by the installer.
Replication Installed	TRUE when components supporting replication are installed.
RPC Encrypt	Whether RPC encryption is enabled.
SNMP	Whether Simple Network Management Protocol (SNMP) is installed on an instance of the SQL Server.
SNMP Current Version	Version of Simple Management Protocol (SNMP) currently installed on an instance of the SQL Server.
Sort Order	Character set used and ordering applied.
SQL Data Root	Default operating system directory implementing storage for SQL Server system user-defined databases.
TCP Port	TCP/IP Sockets Net-Libraries port number on an instance of the SQL Server.

1.1.3 Database Setting Configuration Metrics

The metrics in this category contain the settings for a database. These settings control the access to and the behavior of the database.

- Table Name — MGMT_EMX_MSSQL_DBSETTING
- View Name — MGMT_EMX_MSSQL_DBSETTING_VIEW

Default Collection Interval — Every 24 hours

Table 1–3 Database Setting Configuration Metrics

Metric	Description
Setting ID (key column)	Database name.
SQL ServerName	Name of the SQL Server.
Offline	Whether the database is online. Also, whether the database is unavailable, or is being made unavailable, for use by authorized users.
Recovery Type	Whether the comparison method for multi-byte character data is case-sensitive or not. Type of recovery model that a database will use: Value — 0 Description — Simple Explanation — The database can be recovered only to the last full database backup or last differential backup. Value — 1 Description — Bulk Logged Explanation — Logging for all SELECT INTO, CREATE INDEX, and bulk loading data operations is minimal and therefore requires less log space. In exchange for better performance and less log space usage, the risk of exposure to loss is greater than with full recovery. Value — 2 Description — Full Explanation — Database backups and transaction log backups provide full recoverability from media failure. All operations are fully logged, including bulk operations such as SELECT INTO, CREATE INDEX, and bulk loading data. Value — 3 Description — Unknown Explanation — The recovery type is not known.
AutoClose	Whether the database is closed and its resources are freed when no user connection accesses the database.
AutoCreateStat	Whether the optimizer directs automatic creation of supporting data statistics as required.
AutoShrink	Whether operating system files maintaining table and index data are evaluated for downward resizing when the server periodically checks for unused space.
AutoUpdateStat	Whether the optimizer directs the automatic rebuilding of statistics.
CursorCloseOnCommit	Whether cursors are closed when a transaction is completed.
DataSpaceUsage	Amount of space in use and reserved for use of data in megabytes.
IndexSpaceUsage	Amount of space for the index in megabytes.
DBOUseOnly	Whether only users with the database ownership privilege can access the database.
SingleUser	Whether only one user can access the database at a given time.
ReadOnly	Whether the database is read-only.
DefaultCursor	Whether cursors declared in a batch are created with local scope.
SelectIntoBulkCopy	Whether non-logged operations are allowed.
TruncateLogOnChekpoint	Whether the SQL Server removes log entries referencing committed transactions when activity on the databases forces a dirty page write.

1.2 Access Methods Metrics

The metrics in this category search through and measure the allocation of SQL Server database objects, such as the number of index searches or number of pages that are allocated to indexes and data.

1.2.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 30 minutes

Table 1–4 Access Methods Metrics

Metric	Description
Access Method Counter Name (key column)	Performance metric name. See Table 1–5.
Access Method Counter Value	Performance metric value.

The Access Method Counter Name key column contains several metrics. Table 1–5 provides a list of these metrics and a description for each.

Table 1–5 Access Method Counter Name Metrics

Metric	Description
Extents Deallocations/sec	Number of extents deallocated per second from database objects used for storing index or data records.
Extents Allocated/sec	Number of extents allocated per second to database objects used for storing index or data records.
Forwarded Records/sec	Number of records per second fetched through forwarded record pointers.
FreeSpace Page Fetches/sec	Number of pages returned per second by free space scans used to satisfy requests to insert record fragments.
FreeSpace Scans/sec	Number of scans per second that were initiated to search for free space in which to insert a new record fragment.
Full Scans/sec	Number of unrestricted full scans per second, which can be either base-table or full-index scans.
Index Searches/sec	Number of index searches per second. These are used to start range scans and single index record fetches and to reposition an index.
Mixed Page Allocations/sec	Number of pages allocated per second from mixed extents. These are used for storing the first eight pages that are allocated to an index or table.
Page Deallocations/sec	Number of pages deallocated per second from database objects used for storing index or data records.
Page Splits/sec	Number of page splits per second that occur because of overflowing index pages.
Pages Allocated/sec	Number of pages allocated per second to database objects used for storing index or data records.
Probe Scans/sec	Number of probe scans per second. These are used to directly find rows in an index or base table.
Range Scans/sec	Number of qualified range scans through indexes per second.
Scan Point Revalidations/sec	Number of times per second that the scan point had to be revalidated to continue the scan.
Skipped Ghosted Records/sec	Number of ghosted records per second skipped during scans.
Table Lock Escalations/sec	Number of times locks on a table were escalated.
Workfiles Created/sec	Number of workfiles created per second.

Table 1–5 (Cont.) Access Method Counter Name Metrics

Metric	Description
Worktables Created/sec	Number of work tables created per second.
Worktables from Cache Base	Denominator ("base") of a fraction that the performance counter Worktables from Cache ratio represents.
Worktables from Cache Ratio	Percentage of work tables created where the initial pages were immediately available in the work table cache.

1.2.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 30 minutes

Table 1–6 Access Method Counter Name Metrics

Metric	Description
Extents Deallocations/sec	Number of extents deallocated per second from database objects used for storing index or data records.
Extents Allocated/sec	Number of extents allocated per second to database objects used for storing index or data records.
Forwarded Records/sec	Number of records per second fetched through forwarded record pointers.
FreeSpace Page Fetches/sec	Number of pages returned per second by free space scans used to satisfy requests to insert record fragments.
FreeSpace Scans/sec	Number of scans per second that were initiated to search for free space in which to insert a new record fragment.
Full Scans/sec	Number of unrestricted full scans per second, which can be either base-table or full-index scans.
Index Searches/sec	Number of index searches per second. These are used to start range scans and single index record fetches and to reposition an index.
Mixed Page Allocations/sec	Number of pages allocated per second from mixed extents. These are used for storing the first eight pages that are allocated to an index or table.
Page Deallocations/sec	Number of pages deallocated per second from database objects used for storing index or data records.
Page Splits/sec	Number of page splits per second that occur because of overflowing index pages.
Pages Allocated/sec	Number of pages allocated per second to database objects used for storing index or data records.
Probe Scans/sec	Number of probe scans per second. These are used to directly find rows in an index or base table.
Range Scans/sec	Number of qualified range scans through indexes per second.
Scan Point Revalidations/sec	Number of times per second that the scan point had to be revalidated to continue the scan.
Skipped Ghosted Records/sec	Number of ghosted records per second skipped during scans.
Table Lock Escalations/sec	Number of times locks on a table were escalated.
Workfiles Created/sec	Number of workfiles created per second.
Worktables Created/sec	Number of work tables created per second.
Worktables from Cache Base	Denominator ("base") of a fraction that the performance counter Worktables from Cache ratio represents.
Worktables from Cache Ratio	Percentage of work tables created where the initial pages were immediately available in the work table cache.

1.3 Agent Status Metrics

The metrics in this category provide information regarding the current status of the Agent.

Default Collection Interval — Every 5 minutes

Table 1–7 Agent Status Metrics

Metric	Description and User Action
Process ID	Process ID of the Sqlserver Agent process.
Server name	Name of the Sqlserver instance.
Software Home	Path of the Sqlserver process.
Sqlserver Agent Status	Status of the Sqlserver Agent process. When the status is not running, the SQL server Agent must be started.

1.4 Buffer Manager Metrics

The Buffer Manager object provides counters to monitor how Microsoft SQL Server uses:

- Memory to store data pages, internal data structures, and the procedure cache.
- Counters to monitor the physical I/O as the SQL Server reads database pages from, and writes database pages to, the disk.

1.4.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 15 minutes

Table 1–8 Buffer Manager Metrics

Metric	Description
Buffer Manager Counter Name (key column)	Performance metric name. See Table 1–9.
Buffer Manager Counter Value	Performance metric value.

The Buffer Manager Counter Name key column contains several metrics. Table 1–9 provides a list of these metrics and a description for each.

Table 1–9 Buffer Manager Counter Name Metrics

Metric	Description
Buffer Cache Hit Ratio	Percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since the SQL Server was started. After a long period of time, the ratio does not change very much. Because reading from the cache is much less expensive than reading from disk, this ratio should be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to the SQL Server.
Buffer Cache Hit Ratio Base	Denominator ("base") of a fraction that the performance counter Buffer Cache Hit Ratio represents.
Checkpoint Pages/sec	Number of pages flushed to disk per second by a checkpoint or other operations that cause all dirty pages to be flushed to disk.
Database Pages	Total number of database pages.

Table 1–9 (Cont.) Buffer Manager Counter Name Metrics

Metric	Description
Free List Stalls/sec	Number of requests that had to wait for a free page.
Free Pages	Total number of pages on all free lists.
Lazy Writes/sec	Number of buffers written per second by the buffer manager's lazy writer. The lazy writer is a system process that flushes out batches of dirty, aged buffers (buffers that contain changes that must be written back to disk before the buffer can be reused for a different page) and make them available to user processes. The lazy writer eliminates the need to perform frequent checkpoints in order to create available buffers.
Page Lookups/sec	Number of requests to find a page in the buffer pool.
Page Reads/sec	Number of physical database page reads issued per second. This statistic displays the total number of physical page reads across all databases. Because physical I/O is expensive, you may be able to minimize the cost by using a larger data cache, intelligent indexes, more efficient queries, or by changing the database design.
Page Writes/sec	Number of database page writes issued per second. Page writes are generally expensive. Reducing page-write activity is important for optimal tuning. One way to do this is to ensure that you do not run out of free buffers in the free buffer pool. If you do, page writes will occur while waiting for an unused cache buffer to flush.
Procedure Cache Pages	Number of pages used to store compiled queries.
Readahead Pages/sec	Number of pages read in anticipation of use.
Reserved Pages	Number of buffer pool reserved pages.
Stolen Pages	Number of pages used for miscellaneous server purposes (including procedure cache).
Target Pages	Ideal number of pages in the buffer pool.
Total Pages	Number of pages in the buffer pool (includes database, free, and stolen pages).

1.4.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 15 minutes

Table 1–10 Buffer Manager Counter Name Metrics

Metric	Description
Buffer Cache Hit Ratio	Percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since the SQL Server was started. After a long period of time, the ratio does not change very much. Because reading from the cache is much less expensive than reading from disk, this ratio should be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to the SQL Server.
Buffer Cache Hit Ratio Base	Denominator ("base") of a fraction that the performance counter Buffer Cache Hit Ratio represents.
Checkpoint Pages/sec	Number of pages flushed to disk per second by a checkpoint or other operations that cause all dirty pages to be flushed to disk.
Database Pages	Total number of database pages.
Free List Stalls/sec	Number of requests that had to wait for a free page.
Free Pages	Total number of pages on all free lists.
Lazy Writes/sec	Number of buffers written per second by the buffer manager's lazy writer. The lazy writer is a system process that flushes out batches of dirty, aged buffers (buffers that contain changes that must be written back to disk before the buffer can be reused for a different page) and make them available to user processes. The lazy writer eliminates the need to perform frequent checkpoints in order to create available buffers.
Page Lookups/sec	Number of requests to find a page in the buffer pool.

Table 1–10 (Cont.) Buffer Manager Counter Name Metrics

Metric	Description
Page Reads/sec	Number of physical database page reads issued per second. This statistic displays the total number of physical page reads across all databases. Because physical I/O is expensive, you may be able to minimize the cost by using a larger data cache, intelligent indexes, more efficient queries, or by changing the database design.
Page Writes/sec	Number of database page writes issued per second. Page writes are generally expensive. Reducing page-write activity is important for optimal tuning. One way to do this is to ensure that you do not run out of free buffers in the free buffer pool. If you do, page writes will occur while waiting for an unused cache buffer to flush.
Procedure Cache Pages	Number of pages used to store compiled queries.
Readahead Pages/sec	Number of pages read in anticipation of use.
Reserved Pages	Number of buffer pool reserved pages.
Stolen Pages	Number of pages used for miscellaneous server purposes (including procedure cache).
Target Pages	Ideal number of pages in the buffer pool.
Total Pages	Number of pages in the buffer pool (includes database, free, and stolen pages).

1.5 Cache Manager Metrics

The Cache Manager object provides counters to monitor how the Microsoft SQL Server uses memory to store objects such as stored procedures, ad hoc and prepared Transact-SQL statements, and triggers. Multiple instances of the Cache Manager object can be monitored at the same time, with each instance representing a different type of plan to monitor.

1.5.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 15 minutes

Table 1–11 Cache Manager Metrics

Metric	Description
Cache Manager Counter Name (key column)	Performance metric name. See Table 1–12.
Cache Manager Instance Name (key column)	Instance for the Cache Manager counter name.
Cache Manager Counter Value	Performance metric value.

The Cache Manager Counter Name key column contains several metrics. Table 1–12 provides a list of these metrics and a description for each.

Table 1–12 Cache Manager Counter Name Metrics

Metric	Description
Cache Hit Ratio	Percentage of pages found in the cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since the SQL Server was started. After a long period of time, the ratio does not change very much. Because reading from the cache is less expensive than reading from disk, this ratio should be high. Generally, you can increase the cache hit ratio by increasing the amount of memory available to the SQL Server.
Cache Hit Ratio Base	Denominator ("base") of a fraction that the performance counter Cache Hit Ratio represents.
Cache Pages	Number of pages used by objects in the cache. After a long period of time, the count does not change very much.
Cache Object Counts	Number of objects found in the cache. After a long period of time, the count does not change very much.
Cache Use Counts/sec	Number of times per second that each type of object in the cache has been used. The higher this value is, the better. After a long period of time, the count does not change very much.

1.5.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 15 minutes

Table 1–13 Cache Manager Counter Name Metrics

Metric	Description
Name (key column)	Instance for the Cache Manager counter name.
Cache Hit Ratio	Percentage of pages found in the cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since the SQL Server was started. After a long period of time, the ratio does not change very much. Because reading from the cache is less expensive than reading from disk, this ratio should be high. Generally, you can increase the cache hit ratio by increasing the amount of memory available to the SQL Server.
Cache Hit Ratio Base	Denominator ("base") of a fraction that the performance counter Cache Hit Ratio represents.
Cache Pages	Number of pages used by objects in the cache. After a long period of time, the count does not change very much.
Cache Object Counts	Number of objects found in the cache. After a long period of time, the count does not change very much.
Cache Use Counts/sec	Number of times per second that each type of object in the cache has been used. The higher this value is, the better. After a long period of time, the count does not change very much.

1.6 Database Metrics

The MSSQL_Database class represents a SQL Server database. Each SQL Server installation can contain one or more databases.

Default Collection Interval — Every 15 minutes

Table 1–14 Database Metrics

Metric	Description and User Action
SQL Server Name (key column)	SQL Server name, which is the instance ID specified in the database functional template.
Name (key column)	Database name.
Create Date	Time and date the database was created.
Database File Path	Primary location of the database files.
Database Status	Status of the database: 0 — Normal 32 — Loading 192 — Recovering 256 — Suspect 512 — Offline 1024 — Standby 32768 — Emergency Mode
Size (MB)	Total size of the database in megabytes. Allocate more space to the database if this metric decreases beyond the critical threshold.
Space Available %	Percentage of space that is available. Allocate more space to the database if this metric decreases beyond the critical threshold.
Space Available (KB)	Unused space in kilobytes. Allocate more space to the database if this metric decreases beyond the critical threshold.
Version	Version of Microsoft SQL Server used to create the referenced database.

1.7 Database Backup Metrics

The metrics in this category provide detailed backup information for all databases.

Default Collection Interval — Every 30 minutes

Table 1–15 Database Backup Metrics

Metric	Description
Database_name (key column)	Name of the database.
Name (key column)	Backup name.
Backup Finish Time	Time the backup finished.
Backup Start Time	Time the backup started.
File Location	Physical location of the files.
File Logical Name	Logical name of the files.
File Size	Size of the file.
File Type	D signifies a data file, and L signifies a log file.
Physical Device Name	Name of the physical device.
Unique Backup Set ID	Unique Backup Set UID for the backup.

1.8 Database Information Metrics

The metrics in this category report information about a specified database or all databases.

Default Collection Interval — Every 30 minutes

Table 1–16 Database Information Metrics

Metric	Description
Database_name (key column)	Name of the database.
Compatibility Level	Compatibility level of the database.
Database ID	Unique ID of the database.
Database Owner	Owner of the database.
Database Size	Current size of the database.
Date Created	Creation date of the database.
Status	Status of the database.

1.9 Database Job Metrics

The metrics in this category return information about jobs that are used by the SQLServerAgent service to perform automated activities in Microsoft SQL Server.

Default Collection Interval — Every 15 minutes

Table 1–17 Database Job Metrics

Metric	Description and User Action
job_id (key column)	Job identification number.
Computer Used to Send Network Messages	Name of the user or computer used when sending network messages.
Computer Used to Send Pages	Name of the user or computer used when sending a page.
Current Execution Status	0 — Returns only jobs that are not idle or suspended 1 — Executing 2 — Waiting for thread 3 — Between retries 4 — Idle 5 — Suspended 7 — Performing completion actions
Current Execution Steps in the Job	Current job execution step.
Current Retry Attempt	If the job is running and the step has been retried, this is the current retry attempt.
Delete Job Event	Bitmask indicating under what circumstances the job should be deleted when a job completes. Possible values are the same as for notify_level_eventlog.
Description	Description for the job.
Email of Operator	Email name of the operator to notify.
Enabled	Indicates whether the job is enabled to be executed.
ID of Next Run Schedule	Identification number of the next run schedule.
Job Category	The category to which the job belongs.
Job Creation Date	Date the job was created.

Table 1–17 (Cont.) Database Job Metrics

Metric	Description and User Action
Job Modification Date	Date the job was last modified.
Job Owner	The owner of the job.
Job Type	1 — Local job 2 — Multiserver job 0 — Job has no target servers
Job Version Number	Version of the job, which is automatically updated each time the job is modified.
Last Run Date (mm-dd-yyyy)	Date the job last started executing.
Last Run Outcome	Outcome of the job the last time it ran: 0 — Failed 1 — Succeeded 3 — Canceled 5 — Unknown
Last Run Time (hh:mm:ss)	Time the job last started executing.
Name	Name of the job.
Next Run Date (mm-dd-yyyy)	Date the job is next scheduled to run.
Next Run Time (hh:mm:ss)	Time the job is next scheduled to run.
Notify Level Email	Bitmask indicating under what circumstances a notification email should be sent when a job completes. Possible values are the same as for notify_level_eventlog.
Notify Level Event Log	Bitmask indicating under what circumstances a notification event should be logged to the Microsoft Windows NT application log. Possible values are as follows: 0 — Never 1 — When a job succeeds 2 — When the job fails 3 — Whenever the job completes (regardless of the job outcome)
Notify Level Net Send	Bitmask indicating under what circumstances a network message should be sent when a job completes. Possible values are the same as for notify_level_eventlog.
Notify Level Page	Bitmask indicating under what circumstances a page should be sent when a job completes. Possible values are the same as for notify_level_eventlog.
Number of Job Schedules	Number of job schedules the job has.
Number of Job Steps	Number of job steps the job has.
Number of Target Servers	Number of target servers the job has.
Originating Server	Name of the server from which the job originated.
Start Step ID	ID of the step in the job where execution should begin.

1.10 Database Lock Metrics

The metrics in this category report information about locks.

Default Collection Interval — Every 15 minutes

Table 1–18 Database Lock Metrics

Metric	Description
spid (key column)	Server process ID of the current user process.
dbid (key column)	Database identification number requesting a lock.
ObjId (key column)	Object identification number of the object requesting a lock.
IndID (key column)	The index identification number.
Mode	Lock mode: Shared (S) Update (U) Exclusive (X) Intent Schema Bulk update (BU) RangeS_S — Shared range, shared resource lock; serializable range scan. RangeS_U — Shared range, update resource lock; serializable update scan. RangeI_N — Insert range, null resource lock. Used to test ranges before inserting a new key into an index. RangeX_X — Exclusive range, exclusive resource lock. Used when updating a key in a range.
Resource	Lock resource that corresponds to the value in `syslockinfo.restext`: RID, KEY, PAG, EXT, TAB, and DB
Status	The current status of the lock: GRANT, WAIT, and CNVT

1.11 Database Parameter Metrics

The Databases object in Microsoft SQL Server provides counters to monitor:

- Bulk copy operations.
- Backup and restore throughput.
- Transaction log activities.

Monitoring transactions and the transaction log determine how much user activity is occurring in the database and how full the transaction log is becoming. The amount of user activity can determine the performance of the database and affect log size, locking, and replication. Monitoring low-level log activity to gauge user activity and resource usage can help you identify performance bottlenecks.

Default Collection Interval — Every 24 hours

Table 1–19 Database Parameter Metrics

Metric	Description and User Action
Name (key column)	Name of the Database Configuration parameter.
Current Value	Current value of the Database Configuration parameter.
Description	Text description of the configuration value.
Dynamic Reconfigure	Whether the parameter can be dynamically reconfigured. If TRUE, a modification to the value is immediately effective. If FALSE, modifications are visible only after the SQL Server service has been stopped and restarted.
ID	Parameter name.

Table 1–19 (Cont.) Database Parameter Metrics

Metric	Description and User Action
Maximum Value	Upper bound for a configuration value.
Minimum Value	Lower bound for a configuration value.
Running Value	Value for the configuration option (value in `syscurconfigs.value`).

Table 1–20 ID Description Mapping

Metric	Description
101	Recovery interval.
102	Allow updates.
103	User Connections.
106	Locks.
107	Open objects
109	Fill factor.
115	Nested triggers.
117	Remote access.
124	Default language.
125	Language in cache.
502	Max async I/O.
503	Max worker threads.
505	Network packet size.
518	Show advanced option.
542	Remote proc trans.
543	Remote conn timeout.
1110	Time slice.
1123	Default sort order ID.
1124	Unicode local ID.
1125	Unicode comparison style.
1126	Language neutral.
1127	Two-digit year cutoff.
1505	Index create mem.
1514	Spin Counter.
1517	Priority boost.
1519	Remote login timeout.
1520	Remote query timeout.
1531	Cursor threshold.
1532	Set working set size.
1533	Resource timeout.
1534	User Options.
1535	Processor affinity mask.
1536	Max text repl size.

Table 1–20 (Cont.) ID Description Mapping

Metric	Description
1537	Media retention.
1538	Cost threshold for parallelism.
1539	Max degree of parallelism.
1540	Min memory per query.
1541	Query wait.
1542	VLM size.
1543	Min memory.
1544	Max memory.
1545	Query max time.
1546	Lightweight pooling.

1.12 Database Performance Metrics

The Databases object in Microsoft SQL Server provides counters to monitor:

- Bulk copy operations.
- Backup and restore throughput.
- Transaction log activities.

Monitoring transactions and the transaction log determine how much user activity is occurring in the database and how full the transaction log is becoming. The amount of user activity can determine the performance of the database and affect log size, locking, and replication. Monitoring low-level log activity to gauge user activity and resource usage can help you identify performance bottlenecks.

1.12.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 15 minutes

Table 1–21 Database Performance Metrics

Metric	Description
Database Performance Counter Name (key column)	Performance metric name. See Table 1–22.
Database Performance Instance Name (key column)	Instance for the Database Performance Counter Name
Database Performance Counter Value	Performance metric value.

The Database Performance Counter Name key column contains several metrics. Table 1–22 provides a list of these metrics and a description for each.

Table 1–22 Database Performance Counter Name Metrics

Metric	Description
Active Transactions	Number of active transactions for the database.
Backup/Restore Throughput/sec	Read/write throughput for backup and restore operations of a database per second. For example, you can measure how the performance of the database backup operation changes when more backup devices are used in parallel or when faster devices are used. Throughput of a database backup or restore operation allows you to determine the progress and performance of your backup and restore operations.
Bulk Copy Rows/sec	Number of rows bulk-copied per second.
Bulk Copy Throughput/sec	Amount of data bulk-copied in kilobytes per second.
Data File(s) Size (KB)	Cumulative size (in kilobytes) of all the data files in the database including any automatic growth. Monitoring this counter is useful, for example, for determining the correct size of tempdb.
DBCC Logical Scan Bytes/sec	Number of logical read scan bytes per second for database consistency checker (DBCC) statements.
Log Bytes Flushed/sec	Total number of log bytes flushed.
Log Cache Hit Ratio	Percentage of log cache reads satisfied from the log cache.
Log Cache Reads/sec	Reads performed per second through the log manager cache.
Log File(s) Size	Cumulative size in kilobytes of all the transaction log files in the database.
Log File(s) Used Size (KB)	The cumulative used size of all the log files in the database.
Log Flush Wait Time	Total wait time in milliseconds to flush the log.
Log Flush Waits/sec	Number of commits per second waiting for the log flush.
Log Flushes/sec	Number of log flushes per second.
Log Growths	Total number of times the transaction log for the database has expanded.
Log Shrinks	Total number of times the transaction log for the database has contracted.
Log Truncations	Total number of times the transaction log for the database has truncated.
Percent Log Used	Percentage of space in the log that is in use.
Repl. Pending Xacts	Number of transactions in the transaction log of the publication database marked for replication, but not yet delivered to the distribution database.
Repl. Trans. Rate	Number of transactions per second read out of the transaction log of the publication database and delivered to the distribution database.
Shrink Data Movement Bytes/sec	Amount of data being moved per second by autoshrink operations, DBCC SHRINKDATABASE, or DBCC SHRINKFILE statements.
Transactions/sec	Number of transactions started for the database per second.

1.12.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 15 minutes

Table 1–23 Database Performance Counter Name Metrics

Metric	Description
Name (key column)	Instance for the Database Performance Counter Name
Active Transactions	Number of active transactions for the database.
Backup/Restore Throughput/sec	Read/write throughput for backup and restore operations of a database per second. For example, you can measure how the performance of the database backup operation changes when more backup devices are used in parallel or when faster devices are used. Throughput of a database backup or restore operation allows you to determine the progress and performance of your backup and restore operations.
Bulk Copy Rows/sec	Number of rows bulk-copied per second.
Bulk Copy Throughput/sec	Amount of data bulk-copied in kilobytes per second.
Data File(s) Size (KB)	Cumulative size (in kilobytes) of all the data files in the database including any automatic growth. Monitoring this counter is useful, for example, for determining the correct size of tempdb.
DBCC Logical Scan Bytes/sec	Number of logical read scan bytes per second for database consistency checker (DBCC) statements.
Log Bytes Flushed/sec	Total number of log bytes flushed.
Log Cache Hit Ratio	Percentage of log cache reads satisfied from the log cache.
Log Cache Reads/sec	Reads performed per second through the log manager cache.
Log File(s) Size	Cumulative size in kilobytes of all the transaction log files in the database.
Log File(s) Used Size (KB)	The cumulative used size of all the log files in the database.
Log Flush Wait Time	Total wait time in milliseconds to flush the log.
Log Flush Waits/sec	Number of commits per second waiting for the log flush.
Log Flushes/sec	Number of log flushes per second.
Log Growths	Total number of times the transaction log for the database has expanded.
Log Shrinks	Total number of times the transaction log for the database has contracted.
Log Truncations	Total number of times the transaction log for the database has truncated.
Percent Log Used	Percentage of space in the log that is in use.
Repl. Pending Xacts	Number of transactions in the transaction log of the publication database marked for replication, but not yet delivered to the distribution database.
Repl. Trans. Rate	Number of transactions per second read out of the transaction log of the publication database and delivered to the distribution database.
Shrink Data Movement Bytes/sec	Amount of data being moved per second by autoshrink operations, DBCC SHRINKDATABASE, or DBCC SHRINKFILE statements.
Transactions/sec	Number of transactions started for the database per second.

1.13 Database Session Metrics

The metrics in this category provide information about current Microsoft SQL Server users and processes. The information returned can be filtered to return only the processes that are not idle.

Default Collection Interval — Every 15 minutes

Table 1–24 Database Session Metrics

Metric	Description
spid (key column)	System process ID.
ecid (key column)	Execution context ID of a given thread associated with a specific SPID. ECID = {0, 1, 2, 3, ...n}, where 0 always represents the main or parent thread, and {1, 2, 3, ...n} represent the sub-threads.
Block	System process ID for the blocking process, if one exists. Otherwise, this column is zero. When a transaction associated with a given SPIF is blocked by an orphan distributed transaction, this column returns a -2 for the blocking orphan transaction.
Command	SQL Server command (Transact-SQL statement, SQL Server internal engine process, and so forth) executing for the process.
Database Name	Database used by the process.
Host Name	Host or computer name for each process.
Login Name	Login name associated with the particular process.
Status	Process status.

1.14 Error Log Entry Metrics

The MSSQL_ErrorLogEntry class represents the entries in a SQL Service error log.

1.14.1 For Microsoft SQL Server Database 2000

Default Collection Interval — Every 30 minutes

Table 1–25 Error Log Entry Metrics

Metric	Description
ArchiveID (key column)	Archive number of the log the entry is stored in. The active log has number 0.
EntryID (key column)	Number of the entry within a log. Each entry in a log is successively numbered.
Text	Text message for the corresponding error log entry.

1.14.2 For Microsoft SQL Server Database 2005

Default Collection Interval — Every 60 minutes

Table 1–26 Database Performance Counter Name Metrics

Metric	Description
Type of Record (key column)	Specifies the type of event. This is an enumerated string
Record Number (key column)	Identifies the event within the Windows logfile (for example, NT Eventlog logfile). This is specific to the logfile and is used together with the logfile name to uniquely identify an instance of this class.
Event Log Entry	Name of Windows logfile (for example, NT Eventlog logfile). This is used together with the RecordNumber to uniquely identify an instance of this class.
Source	Variable-length null-terminated string specifying the name of the source (application, service, driver, subsystem) that generated the entry. It is used together with the EventIdentifier to uniquely identify an NT event.
Event Code	This property has the value of the lower 16-bits of the EventIdentifier property. It is present to match the value displayed in the NT Event Viewer. Two events from the same source may have the same value for this property but may have different severity and EventIdentifier values

Table 1–26 (Cont.) Database Performance Counter Name Metrics

Metric	Description
Event Identifier	Identifies the event. This is specific to the source that generated the event log entry, and is used, together with SourceName, to uniquely identify an NT event type.
Date-Time	Date and time of event generation.
Event Severity	Indicates the severity of the event. Two events from the same source may have may have different severity and EventIdentifier values.
Category	Specifies a subcategory for this event. This subcategory is source specific.
User	User name of the logged on user when the event ocurred. If the user name cannot be determined this will be NULL.
Event Message	Event message as it appears in the NT Eventlog. This is a standard message with zero or more insertion strings supplied by the source of the NT event. The insertion strings are inserted into the standard message in a predefined format. If there are no insertion strings or there is a problem inserting the insertion strings, only the standard message will be present in this field.

1.15 General Statistics Metrics

The General Statistics object in Microsoft SQL Server provides counters to monitor general server-wide activity, such as the number of current connections and the number of users connecting and disconnecting per second from computers running an instance of SQL Server. This can be useful when you are working on large online transaction processing (OLTP) systems where many clients connect and disconnect from an instance of SQL Server.

1.15.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 30 minutes

Table 1–27 General Statistics Metrics

Metric	Description
General Statistics Counter Name (key column)	Performance metric name. See Table 1–28.
General Statistics Counter Value	Performance metric value.

The General Statistics Counter Name key column contains several metrics. Table 1–28 provides a list of these metrics and a description for each.

Table 1–28 General Statistics Counter Name Metrics

Metric	Description
Logins/sec	Total number of logins started per second.
Logouts/sec	Total number of logout operations started per second.
User Connections	Number of user connections. Because each user connection consumes some memory, configuring overly high numbers of user connections could affect throughput. User connections should be set to the maximum expected number of concurrent users.

1.15.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 30 minutes

Table 1–29 General Statistics Counter Name Metrics

Metric	Description
Logins/sec	Total number of logins started per second.
Logouts/sec	Total number of logout operations started per second.
User Connections	Number of user connections. Because each user connection consumes some memory, configuring overly high numbers of user connections could affect throughput. User connections should be set to the maximum expected number of concurrent users.

1.16 Integrated Security Setting Metrics

Note: This metric is supported for SQLServer Database 2000 targets but not for SQLServer Database 2005 targets.

The MSSQL_IntegratedSecuritySetting class represents the security settings of a SQL Server installation. This setting affects all login connections to the SQL Server regardless of the login authentication type.

Default Collection Interval — Every 24 hours

Table 1–30 Integrated Security Setting Metrics

Metric	Description and User Action
SettingID (key column)	SQL Server name.
Audit Level	Indicates the current audit level security setting. Possible values are shown below. You can change the value for the desired auditing level. 0 Description — None Explanation — Do not log authentication attempts. 1 Description — Audit Login Success Explanation — Log successful authentication. 2 Description — Audit Login Failure Explanation — Log failed authentication. 3 Description — Audit All Explanation — Log all authentication attempts regardless of success or failure.
Impersonate Client	Indicates the current audit level security setting as shown Security context for non-administrative users executing `xp_cmdshell`. If TRUE, `xp_cmdshell` runs in the security context of the client connection. If FALSE, `xp_cmdshell` runs in the security context of the SQL Server Agent.
Security Mode	Indicates the current security mode. Possible values are shown below. You can change the value for the desired security mode. 0 Description — None Explanation — Do not log authentication attempts. 1 Description — Audit Login Success Explanation — Log successful authentication. 2 Description — Audit Login Failure Explanation — Log failed authentication. 3 Description — Audit All Explanation — Log all authentication attempts regardless of success or failure.

1.17 Last Database Backup Metrics

The metrics in this category provide the last backup information for all databases.

Default Collection Interval — Every 30 minutes

Table 1–31 Last Database Backup Metrics

Metric	Description
database_name (key column)	Name of the database.
Days Since Last Backup	Number of days since the last backup of the database.
Last Backup Date	Date when the last backup of the database was initiated.

1.18 Latches Metrics

The Latches object in the Microsoft SQL Server provides counters to monitor internal SQL Server resource locks called latches. Monitoring the latches to determine user activity and resource usage can help you identify performance bottlenecks.

1.18.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 15 minutes

Table 1–32 Latches Metrics

Metric	Description
Latch Counter Name (key column)	Performance metric name. See Table 1–33
Latch Counter Value	Performance metric value.

The Latch Counter Name key column contains several metrics. Table 1–33 provides a list of these metrics and a description for each.

Table 1–33 Latch Counter Name Metrics

Metric	Description
Average Latch Wait Time (ms)	Average latch wait time in milliseconds for latch requests that had to wait.
Average Latch Wait Time Base	Denominator ("base") of a fraction that the performance counter Average Latch Wait Time represents.
Latch Waits/sec	Number of latch requests that could not be granted immediately and had to wait before being granted.
Total Latch Wait Time	Total latch wait time in milliseconds for latch requests that had to wait in the last second.

1.18.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 15 minutes

Table 1–34 Latch Counter Name Metrics

Metric	Description
Average Latch Wait Time (ms)	Average latch wait time in milliseconds for latch requests that had to wait.

Table 1–34 (Cont.) Latch Counter Name Metrics

Metric	Description
Average Latch Wait Time Base	Denominator ("base") of a fraction that the performance counter Average Latch Wait Time represents.
Latch Waits/sec	Number of latch requests that could not be granted immediately and had to wait before being granted.
Total Latch Wait Time	Total latch wait time in milliseconds for latch requests that had to wait in the last second.

1.19 Login Metrics

The MSSQL_Login class represents the login authentication records present in a SQL Server installation.

Default Collection Interval — Every 5 minutes

Table 1–35 Login Metrics

Metric	Description
Name (key column)	User name.
SQL ServerName (key column)	SQL Server name.
Type	Login type for the user: 0 — Other NT user authentication 1 — NT group 2 — SQL server authentication

1.20 MSSQL Database File Metrics

The MSSQL_DatabaseFile class is an extension to the CIM_DataFile class. It contains properties that are relevant to an operating system file that is also a file storing SQL Server database data.

Default Collection Interval — Every 30 minutes

Table 1–36 MSSQL Database File Metrics

Metric	Description
Name (key column)	User name.
SQL ServerName (key column)	SQL Server name.
Database Name (key column)	Name of the database.
FileGroup Name (key column)	Name of the File Group.
Database File Path	Complete path of the database file.
File Growth	Growth increment of the operating system file that stores table, index, or log data. When FileGrowthType is in megabytes, the FileGrowth value represents the number of megabytes of disk space to allocate for incremental file growth. When FileGrowthType is percent, the value represents a percentage and must be in the range from 1 through 100.
File Growth Type	Method of incremental allocation applied when an operating system file is extended. 0 — Megabyte 1 — Percent 99 — Invalid

Table 1–36 (Cont.) MSSQL Database File Metrics

Metric	Description
Maximum Size	Upper limit for the size of an operating system file containing table and index data, or maintaining a database transaction log.
Primary File	Whether the database file is the one that maintains the database-specific system tables. A SQL Server database can have only one primary file.
Space Available in MB	Amount of disk resources, in megabytes, allocated and unused in operating system files.

1.21 MSSQL File Group Metrics

The MSSQL_FileGroup class represents the groups of operating system files that store a database. A SQL Server filegroup categorizes the operating system files containing data from a single SQL Server database to simplify database administration tasks, such as a backup. A filegroup cannot contain the operating system files of more than one database, though a single database can contain more than one filegroup.

Default Collection Interval — Every 30 hours

Table 1–37 MSSQL File Group Metrics

Metric	Description
DatabaseName (key column)	Name of the database.
Name (key column)	FileGroup name.
SQL ServerName (key column)	Name of the SQL Server.
Default	Whether the filegroup is the default filegroup during table or index creation.
Read Only	Whether the filegroup is read only.
Size	Total size of the file group in megabytes.
Type	Filegroup type. A database is created on exactly one filegroup named PRIMARY. This is the primary filegroup. After database creation, you can add a filegroup to the database, called a user-defined file group. 0 — User-defined 8 — On read-only media 16 — Primary

1.22 MSSQL Transaction Log Metrics

Note: This metric is supported for SQLServer Database 2000 targets but not for SQLServer Database 2005 targets.

The MSSQL_TransactionLog class represents the transaction log of a Microsoft SQL Server database. A SQL Server transaction log maintains a record of modifications to the operating system files containing the data of an SQL Server database. The transaction log provides data recovery assistance if a system failure occurs, and a SQL Server database has at least one operating system file that stores transaction log records. A transaction log can be written to more than one operating system file. Each SQL Server database maintains its own transaction log, and the operating system file or files that store log records cannot be shared with another database.

Default Collection Interval — Every 30 minutes

Table 1–38 MSSQL Transaction Log Metrics

Metric	Description
Name (key column)	FileGroup name.
SQL ServerName (key column)	Name of the SQL Server.
Initial Size	Initial size of the database.
Last Backup	Time of the last backup.
Space Available	Space available in the database.

1.23 Memory Manager Metrics

The Memory Manager object in Microsoft SQL Server provides counters that enable you to monitor overall server memory usage to gauge user activity and resource usage. This can help you identify performance bottlenecks.

1.23.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 15 minutes

Table 1–39 Memory Manager Metrics

Metric	Description
Memory Manager Counter Name (key column)	Performance metric name. See Table 1–40.
Memory Manager Counter Value	Performance metric value.

The Memory Manager Counter Name key column contains several metrics. Table 1–40 provides a list of these metrics and a description for each.

Table 1–40 Memory Manager Counter Name Metrics

Metric	Description
Connection Memory (KB)	Total amount of dynamic memory the server is using for maintaining connections.
Granted Workspace Memory (KB)	Total amount of memory currently granted to executing processes such as hash, sort, bulk copy, and index creation operations.
Lock Memory (KB)	Total amount of dynamic memory the server is using for locks.
Lock Blocks Allocated	Current number of allocated lock blocks. At server startup, the number of allocated lock blocks plus the number of allocated lock owner blocks depends on the SQL Server Locks configuration option. If more lock blocks are needed, the value increases.
Lock Owner Blocks Allocated	Current number of allocated lock owner blocks. At server startup, the number of allocated lock owner blocks plus the number of allocated lock blocks depends on the SQL Server Locks configuration option. If more lock owner blocks are needed, the value increases dynamically.
Lock Blocks	Current number of lock blocks in use on the server (refreshed periodically). A lock block represents an individual locked resource, such as a table, page, or row.
Lock Owner Blocks	Number of lock owner blocks currently in use on the server (refreshed periodically). A lock owner block represents the ownership of a lock on an object by an individual thread. Therefore, if three threads each have a shared (S) lock on a page, there will be three lock owner blocks.

Table 1–40 (Cont.) Memory Manager Counter Name Metrics

Metric	Description
Maximum Workspace Memory (KB)	Maximum amount of memory available for executing processes such as hash, sort, bulk copy, and index creation operations.
Memory Grants Outstanding	Total number of processes per second that have successfully acquired a workspace memory grant.
Memory Grants Pending	Total number of processes per second waiting for a workspace memory grant.
Optimizer Memory (KB)	Total amount of dynamic memory the server is using for query optimization.
SQL Cache Memory (KB)	Total amount of dynamic memory the server is using for the dynamic SQL cache.
Target Server Memory (KB)	Total amount of dynamic memory the server is willing to consume.
Total Server Memory (KB)	The memory allocated to the SQL Server.

1.23.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 15 minutes

Table 1–41 Memory Manager Counter Name Metrics

Metric	Description
Connection Memory (KB)	Total amount of dynamic memory the server is using for maintaining connections.
Granted Workspace Memory (KB)	Total amount of memory currently granted to executing processes such as hash, sort, bulk copy, and index creation operations.
Lock Memory (KB)	Total amount of dynamic memory the server is using for locks.
Lock Blocks Allocated	Current number of allocated lock blocks. At server startup, the number of allocated lock blocks plus the number of allocated lock owner blocks depends on the SQL Server Locks configuration option. If more lock blocks are needed, the value increases.
Lock Owner Blocks Allocated	Current number of allocated lock owner blocks. At server startup, the number of allocated lock owner blocks plus the number of allocated lock blocks depends on the SQL Server Locks configuration option. If more lock owner blocks are needed, the value increases dynamically.
Lock Blocks	Current number of lock blocks in use on the server (refreshed periodically). A lock block represents an individual locked resource, such as a table, page, or row.
Lock Owner Blocks	Number of lock owner blocks currently in use on the server (refreshed periodically). A lock owner block represents the ownership of a lock on an object by an individual thread. Therefore, if three threads each have a shared (S) lock on a page, there will be three lock owner blocks.
Maximum Workspace Memory (KB)	Maximum amount of memory available for executing processes such as hash, sort, bulk copy, and index creation operations.
Memory Grants Outstanding	Total number of processes per second that have successfully acquired a workspace memory grant.
Memory Grants Pending	Total number of processes per second waiting for a workspace memory grant.
Optimizer Memory (KB)	Total amount of dynamic memory the server is using for query optimization.

Table 1–41 (Cont.) Memory Manager Counter Name Metrics

Metric	Description
SQL Cache Memory (KB)	Total amount of dynamic memory the server is using for the dynamic SQL cache.
Target Server Memory (KB)	Total amount of dynamic memory the server is willing to consume.
Total Server Memory (KB)	The memory allocated to the SQL Server.

1.24 Memory Statistics Metrics

The metrics in this category provide information about various memory-related performance issues.

Default Collection Interval — Every 15 minutes

Table 1–42 Memory Statistics Metrics

Metric	Description and User Action
SQL ServerName (key column)	Name of the SQL Server.
Average Latch Wait Time	Average latch wait time in milliseconds for latch requests that had to wait. If this number is high, your server might have resource limitations.
Buffer Cache Hit Ratio	Percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since the SQL Server was started. After a long period of time, the ratio does not change very much. Because reading from the cache is much less expensive than reading from disk, this ratio should be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to the SQL Server.
Cache Hit Ratio	Percentage of pages found in the cache without needing to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since the SQL Server was started. After a long period of time, the ratio does not change very much. Because reading from the cache is less expensive than reading from disk, this ratio should be high. The higher this value is, the better. Generally, you can increase the cache hit ratio by increasing the amount of memory available to the SQL Server.
Log Flush Wait Time	Log cache is very important, because it rolls back a transaction before it is committed if the circumstances warrant. But after a transaction is complete (and no longer can be rolled back), this log cache is immediately flushed to the physical log file. This is a normal procedure. SELECT queries that do not modify data do not create transactions and do not produce log flushes. Essentially, a log flush occurs when data is written from the log cache to the physical log file. Therefore, a log flush occurs every time a transaction completes, and the number of log flushes that occur are related to the number of transactions performed by the SQL Server. One way to troubleshoot the disk I/O bottleneck is to capture the Log Flushes/sec counter data and see how busy this mechanism is. If the server experiences a lot of transactions, it will also experience a lot of log flushes, so the value you see for this counter can vary from server to server, depending on how busy it is with action-type queries that create transactions. Try to identify situations where the number of log flushes per second seems to be significantly higher than the expected number of transactions that you think should be running on a server.
Total Lock Wait Time	Total wait time in milliseconds for locks in the last second. If the value is high, your server has high resource contention.

1.25 Processor Metrics

The Win32_Processor class represents a device that is capable of interpreting a sequence of machine instructions on a Win32 computer system. On a multiprocessor machine, one instance of this class exists for each processor.

Default Collection Interval — Every 15 minutes

Table 1–43 Processor Metrics

Metric	Description and User Action
CPU Status (key column)	Device ID for the device.
CPU Status	Status of the CPU.
Load Percentage	Usage of the CPU. If the value increases above the critical threshold, this indicates a possible risk to the processor.

1.26 Response Metrics

This metrics category provide information about the response of the target Sqlserver Instance.

Default Collection Interval — Every 5 minutes

Table 1–44 Response Metrics

Metric	Description and User Action
Process ID	Process ID of the Sqlserver process.
Server Name	Name of the instance of the Sqlserver.
Software Home	Path of the Sqlserver process.
Sqlserver Status	Status of the Sqlserver process. When the status is not Running, the SQL Server must be started.

1.27 SQL Server Locks Metrics

The Locks object in the Microsoft SQL Server provides information about SQL Server locks on individual resource types. Locks are held on SQL Server resources, such as rows read or modified during a transaction, to prevent concurrent use of resources by multiple transactions. For example, if an exclusive (X) lock is held on a row within a table by a transaction, no other transaction can modify that row until the lock is released. Minimizing locks increases concurrency, which can improve performance. Multiple instances of the Locks object can be monitored at the same time, with each instance representing a lock on a resource type.

1.27.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 15 minutes

Table 1–45 SQL Server Locks Metrics

Metric	Description
SQL Server Locks Counter Name (key column)	Performance metric name. See Table 1–46.
SQL Server Locks Instance Name (key column)	Instance for the SQL Server Locks Counter Name.
SQL Server Locks Counter Value	Performance metric value.

The SQL Server Locks Counter Name key column contains several metrics. Table 1–46 provides a list of these metrics and a description for each.

Table 1–46 SQL Server Locks Counter Name Metrics

Metric	Description
Average Wait Time (ms)	Average amount of wait time in milliseconds for each lock request that resulted in a wait.
Average Wait Time Base	Denominator ("base") of a fraction that the performance counter Average Wait Time ratio represents.
Lock Requests/sec	Number of new locks and lock conversions per second requested from the lock manager.
Lock Timeouts/sec	Number of lock requests per second that timed out, including internal requests for NOWAIT locks.
Lock Waits/sec	Number of lock requests per second that could not be satisfied immediately and required the caller to wait.
Lock Wait Time (ms)	Total wait time in milliseconds for locks in the last second.
Number of Deadlocks/sec	Number of lock requests per second that resulted in a deadlock.

1.27.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 15 minutes

Table 1–47 SQL Server Locks Counter Name Metrics

Metric	Description
Name (key column)	Instance for the SQL Server Locks Counter Name.
Average Wait Time (ms)	Average amount of wait time in milliseconds for each lock request that resulted in a wait.
Average Wait Time Base	Denominator ("base") of a fraction that the performance counter Average Wait Time ratio represents.
Lock Requests/sec	Number of new locks and lock conversions per second requested from the lock manager.
Lock Timeouts/sec	Number of lock requests per second that timed out, including internal requests for NOWAIT locks.
Lock Waits/sec	Number of lock requests per second that could not be satisfied immediately and required the caller to wait.
Lock Wait Time (ms)	Total wait time in milliseconds for locks in the last second.
Number of Deadlocks/sec	Number of lock requests per second that resulted in a deadlock.

1.28 SQL Server Process Metrics

The MSSQL_Process class represents SQL Server processes. Note that these are not the same as an operating system's notion of a process. These are the processes identified by the SQL Server and assigned a SQL Server process ID by the SQL Server.

Default Collection Interval — Every 15 minutes

Table 1–48 SQL Server Process Metrics

Metric	Description and User Action
SQL ServerName (key column)	SQL Server name.
CPU Time (key column)	Process ID.
Blocked Process ID	ID of a process being blocked by the process.
CPU Time	Cumulative CPU usage time of the process.
Client Name	Name of the client application.
Command	Abbreviated indicator of the current command. When no command is current, it has a value of AWAITING COMMAND.
Creation Date	Time that the process began executing.
Database Name	Database currently being used by the process.
Execution State	Current operating condition of the process. Possible values are as shown: 0 — Unknown 1 — Other 2 — Ready 3 — Running 4 — Blocked 5 — Suspended Blocked 6 — Suspended Ready
Host Name	Name of the client workstation that started the SQL Server process.
Kernel Mode Time	Time in kernel mode in milliseconds. If this information is not available, a value of 0 should be used.
Login	Login used by the process to connect to the SQL Server.
Memory Usage	Number of pages in the procedure cache that are currently allocated to this process. A negative number indicates that the process is freeing memory allocated by another process.
Priority	If a priority is not defined for a process, a value of 0 should be used.
State	Whether the process is running or sleeping.
Termination Date	Time that the process was stopped or terminated.
User Mode Time	Number of milliseconds in user mode. If this information is not available, a value of 0 should be used.
Working Set Size	Amount of memory in bytes that a process needs to execute efficiently for an operating system that uses page-based memory management. If a sufficient amount of memory is unavailable (< working set size), thrashing occurs. If this information is not known, NULL or 0 should be entered. If this data is provided, it could be monitored to understand a process' changing memory requirements as execution proceeds.

1.29 SQL Server Role Metrics

The MSSQL_Role class represents a database role or a SQL Server role. Roles establish groups of users with similar security attributes. Permissions can be granted by role, simplifying security planning and administration.

Default Collection Interval — Every 30 minutes

Table 1–49 SQL Server Role Metrics

Metric	Description
Name	Role name.
SQL ServerName	Name of the SQL Server.
Full Name	Descriptive title for the role.

1.30 SQL Statistics Metrics

The SQL Statistics object in the Microsoft SQL Server provides counters to monitor compilation and the type of requests sent to an instance of the SQL Server. Monitoring the number of query compilations and recompilations and the number of batches received by an instance of the SQL Server indicates how quickly the SQL Server is processing user queries and how effectively the query optimizer is processing the queries.

1.30.1 For Microsoft SQL Server 2000

Default Collection Interval — Every 10 minutes

Table 1–50 SQL Statistics Metrics

Metric	Description
SQL Statistics Counter Name (key column)	Performance metric name. See Table 1–51.
SQL Statistics Counter Value (key column)	Performance metric value.

The SQL Statistics Counter Name key column contains several metrics. Table 1–51 provides a list of these metrics and a description for each.

Table 1–51 SQL Statistics Counter Name Metrics

Metric	Description
Auto-Param Attempts/sec	Number of auto-parameterization attempts per second. Total should be the sum of the failed, safe, and unsafe auto-parameterizations. Auto-parameterization occurs when the SQL Server attempts to reuse a cached plan for a previously executed query that is similar as the current query, but not exactly the same. For more information, see "Auto-parameterization" in the Microsoft SQL Server Introduction.
Batch Requests/sec	Number of Transact-SQL command batches received per second. This statistic is affected by all constraints (such as I/O, number of users, cache size, complexity of requests, and so forth). High batch requests mean good throughput. For more information, see "Batch Processing" in the Microsoft SQL Server Introduction.
Safe Auto-Params/sec	Number of safe auto-parameterization attempts per second.
SQL Compilations/sec	Number of SQL compilations per second. Indicates the number of times the compile code path is entered. Includes compiles due to recompiles. After SQL Server user activity is stable, this value should reach a steady state.
SQL Recompilations/sec	Number of SQL recompiles per second. Counts the number of times recompiles are triggered. Generally, the number of recompiles should be low.
Unsafe Auto-Params/sec	Number of unsafe auto-parameterization attempts per second. The table has characteristics that prevent the cached plan from being shared. These are designated as unsafe. The fewer of these that occur the better.

1.30.2 For Microsoft SQL Server 2005

Default Collection Interval — Every 10 minutes

Table 1–52 SQL Statistics Counter Name Metrics

Metric	Description
Auto-Param Attempts/sec	Number of auto-parameterization attempts per second. Total should be the sum of the failed, safe, and unsafe auto-parameterizations. Auto-parameterization occurs when the SQL Server attempts to reuse a cached plan for a previously executed query that is similar as the current query, but not exactly the same. For more information, see "Auto-parameterization" in the Microsoft SQL Server Introduction.
Batch Requests/sec	Number of Transact-SQL command batches received per second. This statistic is affected by all constraints (such as I/O, number of users, cache size, complexity of requests, and so forth). High batch requests mean good throughput. For more information, see "Batch Processing" in the Microsoft SQL Server Introduction.
Safe Auto-Params/sec	Number of safe auto-parameterization attempts per second.
SQL Compilations/sec	Number of SQL compilations per second. Indicates the number of times the compile code path is entered. Includes compiles due to recompiles. After SQL Server user activity is stable, this value should reach a steady state.
SQL Recompilations/sec	Number of SQL recompiles per second. Counts the number of times recompiles are triggered. Generally, the number of recompiles should be low.
Unsafe Auto-Params/sec	Number of unsafe auto-parameterization attempts per second. The table has characteristics that prevent the cached plan from being shared. These are designated as unsafe. The fewer of these that occur the better.

1.31 Server Alerts Metrics

The metrics in this category provide details of alerts from the server.

Default Collection Interval — Every 15 minutes

Table 1–53 Server Alerts Metrics

Metric	Description
ID (key column)	System-assigned unique integer identifier.
Category Name	Name of the alert category.
Count Reset Date	Date the `occurrence_count` was last set.
Count Reset Time	Time the `occurrence_count` was last reset.
Database Name	Database in which the error must occur for the alert to appear. If the database name is NULL, the alert appears regardless of where the error occurred.
Delay Between Responses	Seconds of waiting between responses to the alert.
Enabled	Status of whether the alert is currently enabled (1) or not (0). An unenabled alert is not sent.
Event Category ID	Reserved.
Event Description Keyword	Description of the SQL Server error in the Windows NT application log that must be like the supplied sequence of characters.
Event ID	Reserved.
Event Source	Source of the event. It is always MSSQLServer for Microsoft SQL Server version 7.0.
Flags	If type is 2, this column shows the definition of the performance condition; otherwise, the column is NULL.

Table 1–53 (Cont.) Server Alerts Metrics

Metric	Description
Has Notification	Nonzero if one or more operators are notified for this alert. The value is one or more of the following values (ORed together): 1 — Has email notification 2 — Has pager notification 4 — Has netsend notification
Include Event Description	Whether the description of the SQL Server error from the Microsoft Windows NT application log should be included as part of the notification message.
Job ID	Job identification number.
Job Name	An on-demand job to be executed in response to an alert.
Last Occurrence Date	Date the alert last occurred.
Last Occurrence Time	Time the alert last occurred.
Last Response Date	Date the SQLServerAgent service last responded to the alert.
Last Response Time	Time the SQLServerAgent service last responded to the alert.
Message ID	Message error number that defines the alert. (This usually corresponds to an error number in the sysmessages table.) If severity is used to define the alert, `message_id` is 0 or NULL.
Name	Alert name, such as Demo: Full msdb log.
Notification Message	Optional additional message sent to the operator as part of the email or pager notification.
Occurrence Count	Number of times the alert occurred.
Performance Condition	Reserved. Will always be '[uncategorized]' for SQL Server 7.0.
Severity	Severity level (from 9 through 25, 110, 120, 130, or 140) that defines the alert.
Type	1 — SQL Server event alert 2 — SQL Server performance alert

1.32 Server Statistics Metrics

The metrics in this category provide information about various server-related performance issues.

Default Collection Interval — Every 15 minutes

Table 1–54 Server Statistics Metrics

Metric	Description
CPU Busy Ratio	CPU utilization.
CPU ms	CPU busy time in milliseconds.
Errors / sec.	Packet error rate in seconds.
IDLE ms	CPU idle time in milliseconds.
IO ms	IO busy time in milliseconds.
Max Connections	Maximum number of connections.
Open Transactions	Total number of transactions.
Packet Error Ratio	The ratio of erroneous packets received to the number of packets received.
Packets Errors	Number of packet errors.
Packets Received	Number of received packets.
Packets Sent	Number of sent packets.

Table 1–54 (Cont.) Server Statistics Metrics

Metric	Description
Reads / sec.	Packet read rate in seconds.
Total Errors	Total number of errors.
Total Reads	Total number of reads.
Total Writes	Total number of writes.
Writes / sec.	Packet write rate in seconds.

1.33 User Metrics

The User object exposes the attributes of a single Microsoft SQL Server database user.

Default Collection Interval — Every 24 hours

Table 1–55 User Metrics

Metric	Description and User Action
DatabaseName (key column)	Name of the database.
Name (key column)	User name.
SQL ServerName (key column)	Name of the SQL Server.
Status	The status property is a string indicating the current status of the object. Various operational and non-operational statuses can be defined. Operational statuses are OK, Degraded, and Pred Fail. Pred Fail indicates that an element may be functioning properly but predicting a failure in the near future.
System Object	The SystemObject property indicates whether the object is owned by Microsoft. A value of True indicates that the object implementation is owned by Microsoft.

2

IBM DB2 Database Metrics

This chapter provides descriptions for all IBM DB2 Database 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. Shaded rows represent key columns for a particular category.

2.1 Connections

Connections metrics provide the connection details for the database at the Agent level and at the Database Manager level. Connections metrics consist of the following categories:

- Agent Collection Statistics
- Connected Applications Statistics

2.1.1 Agent Connection Statistics Metrics

The metrics in this category return the Agent connection statistics, including the configured values and the current parameter values at the Database Manager snapshot level.

Default Collection Interval — Every 15 minutes

Table 2–1 Agent Connection Statistics Metrics

Metric	Description and User Action
Agent Created Due to Empty Pool	Number of Agents created due to an empty pool.
Agent Creation Rate	Creation rate for Agents in the last interval, which equals: `(agents_created_empty_pool-agents_created_empty_pool)/ __interval` Increase the value of the configuration parameter `agents_created_empty_pool`.
Agent Creation to Assignment Ratio	Creation to assignment ratio, which equals: `(agents_created_empty_pool /agents_from_pool)` Increase the value of the configuration parameter `agents_from_pool`.
Agent Limit	Number of Agents the Database Manager can create.
Agent Waiting Top	Maximum number of Agents that have ever waited.
Agents Assignment Rate	Assignment rate for Agents in the last interval, which equals: `(agents_from_pool-agents_from_pool) /__interval` Increase the value of the configuration parameter `agents_from_pool`.
Agents From Pool	Number of Agents from the pool.
Agents Registered	Number of registered Agents.

Table 2–1 (Cont.) Agent Connection Statistics Metrics

Metric	Description and User Action
Agents Waiting on Token	Number of Agents waiting on a token.
Coordination Agent Limit	Maximum number of coordination Agents the Database Manager can create.
Idle Agents	Number of idle Agents.
Maximum Agents Registered	Maximum number of registered Agents.
Maximum Coordination Agents	Maximum number of coordination Agents the database has created.

2.1.2 Connected Applications Statistics Metrics

The metrics in this category return the current connection information at both the database and Database Manager snapshot level, including the number of active connections.

Default Collection Interval — Every 15 minutes

Table 2–2 Connected Applications Statistics Metrics

Metric	Description
Database Name (key column)	Name of the database
Remote Connections	Current number of connections initiated from remote clients to the instance of the database manager that is being monitored.
Local Connections	Number of local applications currently connected to a database within the database instance being monitored.
Remote Connections in Execution	Number of remote applications currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.
Local Connections in Execution	Number of local applications currently connected to a database that are currently processing a unit of work within the database manager instance being monitored.
Connects Since Database Activation	Number of connections to the database since the first connect, activate, or last reset (coordinator Agents).
Applications Connected Currently	Number of applications currently connected to the database.
Applications Executing in the Database	Number of applications currently connected to the database, and for which the database manager is currently processing a request.

2.2 Database Manager Configuration Information

Database Manager Configuration Information metrics describe a set of Database Manager configuration parameters. These values are of two types: in hard disk and in memory, which is the type presented here. Database Manager Configuration Information metrics consist of the following categories:

- Capacity
- Logging and Recovery
- Connection
- Partitioned Database Environment
- Database Instance

2.2.1 Capacity Metrics

The metrics in this category return the Database Manager capacity configuration parameters that can impact the throughput on your system.

- Table Name — MGMT_EMX_IBMDB2_DBMCAP
- View Name — MGMT_EMX_IBMDB2_DBMCAP_VIEW

Default Collection Interval — Every 24 hours

Table 2–3 Capacity Metrics

Metric	Description
Application Support Layer Size	The application support layer heap represents a communication buffer between the local application and its associated Agent.
Database System Monitor Heap Size	Determines the amount of memory, in pages, to allocate for database monitor data.
Audit Buffer Size	Specifies the size of the buffer used when auditing the database.
Maximum Java Interpreter Heap Size	Determines the maximum size of the heap used by the Java interpreter started to service Java DB2 stored procedures and UDFs.
Maximum Total Files Open	Defines the maximum number of files that can be opened by all Agents and other threads executing in a single database manager instance.
Priority of Agents	Controls the priority the operating system scheduler gives to all Agents and other database manager instance processes and threads.
Maximum Number of Agents	Indicates the maximum number of database manager Agents, whether coordinator Agents or sub-Agents, available at any given time to accept application requests.
Maximum Number of Concurrent Agents	Maximum number of database manager Agents that can concurrently execute a database manager transaction.
Agent Pool Size	Determines the maximum size of the idle Agent pool.
Initial Number of Agents in Pool	Determines the initial number of idle Agents created in the Agentpool at DB2START time.
Sort Heap Threshold	The size of the shared sort memory is statically predetermined at the time of the first connection to a database based on the value of `sheapthreas`.

2.2.2 Connection Metrics

The metrics in this category return the parameters that provide information about using DB2 in a client/server environment.

- Table Name — MGMT_EMX_IBMDB2_DBMCON
- View Name — MGMT_EMX_IBMDB2_DBMCON_VIEW

Default Collection Interval — Every 24 hours

Table 2–4 Connection Metrics

Metric	Description
TCP/IP Service Name	Contains the name of the TCP/IP port that a database server uses to communicate to the client.
Search Discovery Communications Protocols	From an administration server perspective, this metric defines the search discovery managers started when DB2ADMIN starts.

2.2.3 Database Instance Metrics

The metrics in this category return the parameters that provide information about Database Manager instances.

- Table Name — MGMT_EMX_IBMDB2_DBMDBINST
- View Name — MGMT_EMX_IBMDB2_DBMDBINST_VIEW

Default Collection Interval — Every 24 hours

Table 2–5 Database Instance Metrics

Metric	Description
Diagnostic Error Capture Level	Determines the type of diagnostic errors recorded in the `db2diag.log`.
Diagnostic Directory Data Path	Enables you to specify the fully qualified path for DB2 diagnostic information.
Notify Level Raw	Specifies the type of administration notification messages written to the administration notification log.
Default Database System Monitor Switches	Unique metric that enables you to set several switches, each of which are internally represented by a bit of the metric.
Communications Bandwidth	Value calculated for the communications bandwidth in MB per second.
CPU Speed Raw	The CPU speed, in milliseconds per instruction, used by the SQL optimizer to estimate the cost of performing certain operations.
Maximum Number of Concurrently Active Databases	Specifies the number of local databases that can be concurrently active.
System Administration Authority Group Name	Defines the group name with SYSADM authority for the database manager instance.
Notify Level	If `notifylevel_raw` equals: 0 — No messages 1 — Fatal or unrecoverable errors 2 — All Immediate action required messages 3 — All Important information (no immediate action required) Otherwise, All Informational messages.
CPU Speed	CPU speed in MIPs, which equals: `1/(cpuspeed_raw*1000)`

2.2.4 Logging and Recovery Metrics

The metrics in this category save the logging and recovery information. Recovering your environment can be very important to prevent the loss of critical data. A number of parameters are available to help you manage your environment and to ensure that you can adequately recover your data or transactions.

- Table Name — MGMT_EMX_IBMDB2_DBMLOGREC
- View Name — MGMT_EMX_IBMDB2_DBMLOGREC_VIEW

Default Collection Interval — Every 24 hours

Table 2–6 Logging and Recovery Metrics

Metric	Description
Transaction Manager Database Name	Identifies the name of the transaction manager (TM) database for each DB2 instance.
Transaction Resync Interval	Specifies the time interval in seconds for which a transaction manager (TM), resource manager (RM), or sync point manager (SPM) should retry the recovery of any outstanding transactions in doubt found in the TM, RM, or SPM.
Sync Point Manager Name	Identifies the name of the sync point manager (SPM) instance to the database manager.
Sync Point Manager Log File Size	Identifies the sync point manager (SPM) log file size in 4 KB pages.
Sync Point Manager Resync Agent Limit	Identifies the number of Agents that can simultaneously perform resync operations.

2.2.5 Partitioned Database Environment Metrics

The metrics in this category return parameters about parallel operations and partitioned database environments.

- Table Name — MGMT_EMX_IBMDB2_DBMPARENV
- View Name — MGMT_EMX_IBMDB2_DBMPARENV_VIEW

Default Collection Interval — Every 24 hours

Table 2–7 Partitioned Database Environment Metrics

Metric	Description
Connection Elapsed Time	Specifies the number of seconds within which a TCP/IP connection is to be established between two database partition servers.
Number of FCM Buffers	Specifies the number of 4 KB buffers used for internal communications (messages) both among and within database servers.
Node Connection Retries	`max_connretries` specifies the number of connection retries that can be made to a database partition server.
Maximum Time Difference Among Nodes	Each database partition server has its own system clock. This metric specifies the maximum time difference, in minutes, that is permitted among the database partition servers listed in the node configuration file.
Start and Stop Timeout	Applicable only in a partitioned database environment.

2.3 Health Indicators/Alarms

Health Indicators/Alarms metrics return the health information and current values for all the snapshot levels of containers, tablespaces, databases and the Database Manager. Health Indicators/Alarms metrics consist of the following categories:

- Container Health Indicator
- Container Health Information
- Database Health Indicator
- Database Health Information
- Database Collection Health Indicator
- DBM Health Indicator
- DBM Health Information

- Tablespaces Health Indicator
- Tablespaces Health Indicator History

2.3.1 Container Health Indicator Metrics

The metrics in this category return health indicator information for tablespace containers from a health snapshot of tablespaces in a database.

Default Collection Interval — Every 30 minutes

Table 2–8 Container Health Indicator Metrics

Metric	Description and User Action
Container Name	Name of the container.
Health Indicator Alert State	State of the alert. If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm A warning or alarm condition indicates that you should examine the Health Indicator Alert Type.
Health Indicator Alert State Raw	Severity of the alert.
Health Indicator Alert Type	Type of alert. If `alert_state_raw` equals: 3001 — Tablespace Container State 3002 — Tablespace Container Utilization
Health Indicator Identifier	Identifier for the alert.
Health Indicator Timestamp	Time when the alert was generated.
Health Indicator Value	Value for the alert.
Node Number	Node at which the alert was generated.
Snapshot Timestamp	Time when the query was executed.

2.3.2 Container Health Information Metrics

The metrics in this category return container information from a health snapshot of a database.

Default Collection Interval — Every 30 minutes

Table 2–9 Container Health Information Metrics

Metric	Description and User Action
Container Name (key column)	Name of the container.
Tablespace Name (key column)	Name of the tablespace to which the container belongs.
Node Number	Node at which the container resides.

Table 2–9 (Cont.) Container Health Information Metrics

Metric	Description and User Action
Rolled Up Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm
Rolled Up Alert State Raw	Severity of the alert.
Snapshot Timestamp	Time when the query was executed.

2.3.3 Database Collection Health Indicator Metrics

The metrics in this category return container information from a health snapshot of a database.

Default Collection Interval — Every 30 minutes

Table 2–10 Database Collection Health Indicator Metrics

Metric	Description and User Action
Database Name (key column)	Name of the database.
Health Indicator Detail	Description of the object.
Health Indicator ID	Identifier for the alert.
Health Indicator Object Name	Name of the object.
Health Indicator Object State Detail	Type of alert state. A warning or alarm condition indicates that the Health Indicator Alert Type should be examined.
Health Indicator Object State	Severity of the alert.
Health Indicator Timestamp	Time when the alert was generated.
Snapshot Timestamp	Time when the query was executed.

2.3.4 Database Health Indicator Metrics

The metrics in this category return health indicator information from a health snapshot of a database.

Default Collection Interval — Every 30 minutes

Table 2–11 Database Health Indicator Metrics

Metric	Description and User Action
Database Name (key column)	Name of the database.
Health Indicator Alert State Raw	Severity of the alert.
Health Indicator Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm A warning or alarm condition indicates that the Health Indicator Alert Type should be examined.
Health Indicator Alert Type	If `alert_state_raw` equals: 3001 — Tablespace Container State 3002 — Tablespace Container Utilization
Health Indicator ID	Identifier for the alert.
Health Indicator Timestamp	Time when the alert was generated.
Health Indicator Value	Value for the alert.
Snapshot Timestamp	Time when the query was executed.

2.3.5 Database Health Information Metrics

The metrics in this category return information from a health snapshot of a database.

Default Collection Interval — Every 30 minutes

Table 2–12 Database Health Information Metrics

Metric	Description and User Action
Database Name (key column)	Name of the database.
Database Path	Physical path of the database.
Database Location	Location of the database. If `db_location_raw` equals: 0 — Local 1 — Remote
Database Location Raw	Location of the database with respect to the DBM.
Health Indicator Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm A warning or alarm condition indicates there are one or more alerts on the database.
Health Indicator Alert State Raw	Severity of the alert.
Input Database Alias	Alias name for the database.
Server Platform	Platform where the database is installed.
Snapshot Timestamp	Time when the query was executed.

2.3.6 DBM Health Indicator Metrics

The metrics in this category return health indicator information from a health snapshot of the DB2 Database Manager.

Default Collection Interval — Every 30 minutes

Table 2–13 DBM Health Indicator Metrics

Metric	Description and User Action
Database Name (key column)	Name of the database.
Health Indicator Alert State Raw	Severity of the alert.
Health Indicator Alert Type	If `alert_state_raw` equals: 3001 — Tablespace Container State 3002 — Tablespace Container Utilization
Health Indicator Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm A warning or alarm condition indicates that the Health Indicator Alert Type should be examined.
Health Indicator ID	Identifier for the alert.
Health Indicator Timestamp	Time when the alert was generated.
Health Indicator Value	Value for the alert.
Server Instance Name	Host name where DB2 is installed.
Snapshot Timestamp	Time when the query was executed.

2.3.7 DBM Health Information Metrics

The metrics in this category return information from a health snapshot of the DB2 Database Manager.

Default Collection Interval — Every 30 minutes

Table 2–14 DBM Health Information Metrics

Metric	Description and User Action
Server Instance Name (key column)	Host name where DB2 is installed.
Database Start Up Time	Time DB2 was last started.
Database Last Reset Time	Time DB2 was last reset.
Number of Nodes in DB2 Instance	Number of nodes in the DB2 instance.
Rolled Up Alert State Raw	Host name where DB2 is installed.

Table 2–14 (Cont.) DBM Health Information Metrics

Metric	Description and User Action
Rolled Up Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm A warning or alarm condition indicates there are one or more alerts on the DBM.
Server Instance Name	Host name where DB2 is installed.
Snapshot Timestamp	Time when the query was executed.

2.3.8 Tablespaces Health Indicator

The metrics in this category return health indicator information for tablespaces from a health snapshot of tablespaces in a database.

Default Collection Interval — Every 30 minutes

Table 2–15 Tablespaces Health Indicator Metrics

Metric	Description and User Action
Tablespace Name (key column)	Name of the tablespace.
Health Indicator Alert State Raw	Severity of the alert.
Health Indicator Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm
Health Indicator Alert Type	If `alert_state_raw` equals: 3001 — Tablespace Container State 3002 — Tablespace Container Utilization
Health Indicator ID	Identifier for the alert.
Health Indicator Timestamp	Time when the alert was generated.
Health Indicator Value	Value for the alert.
Snapshot Timestamp	Time when the query was executed.

2.3.9 Tablespaces Health Indicator History

The metrics in this category return health indicator information for tablespaces from a health snapshot of tablespaces in a database.

Default Collection Interval — Every 30 minutes

Table 2–16 Tablespaces Health Indicator History Metrics

Metric	Description and User Action
Tablespace Name (key column)	Name of the tablespace.
Rolled Up Alert State Raw	Severity of the alert.
Rolled Up Alert State	If `alert_state_raw` equals: 1 — Normal 2 — Attention 3 — Warning 4 — Alarm
Snapshot Timestamp	Time when the query was executed.

2.4 Monitoring Information

Monitoring Information metrics capture the monitoring information for the database, including general monitoring information, monitored values of the Agent, and monitored values of the database. Monitoring Information metrics consist of the following categories:

- General Information
- Agent Monitoring
- Database Monitoring
- Database Backup

2.4.1 Agent Monitoring Metrics

The metrics in this category return information about Agents from an application snapshot.

Default Collection Interval — Every 15 minutes

Table 2–17 Agent Monitoring Metrics

Metric	Description and User Action
Agent CPU Utilization (%)	Total CPU utilization, which is equal to: `Agent_total_cpu_time-_Agent_total_cpu_time )/1000)/__interval)*100`
Agent Identifier	Unique ID for each Agent.
Agent System CPU Time	Total system time used by DBM for processing by the Agent if it is an application-level snapshot, or by a statement if it is at the statement level.
Agent User CPU Time	Total user time used by DBM for processing by the Agent if it is an application-level snapshot, or by a statement if it is at the statement level.
Application Average Lock Wait Time (ms)	The average waiting time for locks, which equals: `lock_wait_time / lock_wait` If the average lock wait time is high, you should look for applications that hold many locks, or have lock escalations, with a focus on tuning your applications to improve concurrency, if appropriate. If escalations are causing a high average lock wait time, the values of one or both of the locklist and maxlocks configuration parameters may be too low.
Application Commit SQL Statements Rate	Commit SQL statements reading rate, which equals: `commit_sql_stmts - _commit_sql_stmts/ __interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.

Table 2–17 (Cont.) Agent Monitoring Metrics

Metric	Description and User Action
Application Dynamic SQL Statements Rate	Dynamic SQL statements reading rate, which equals: `dynamic_sql_stmts - _dynamic_sql_stmts/ __interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Application Failed SQL Statements Rate	Failed SQL statements reading rate, which equals: `failed_sql_stmts - _failed_sql_stmts/ __interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Application Identifier	Unique ID for each application.
Application Idle Time	Time spent idle for an application.
Application Name	Name of the application.
Application Rollback SQL Statements Rate	Rollback SQL statements reading rate, which equals: `rollback_sql_stmts - _rollback_sql_stmts/ __interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Application Row Reading Rate	Rows reading rate in the last interval, which equals: `rows_read - rows_read)/__interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Application Row Writing Rate	Rows writing rate in the last interval, which equals: `rows_written - rows_written)/__interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Application Static SQL Statements Rate	Static SQL statements reading rate, which equals: `static_sql_stmts - _static_sql_stmts/ __interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Application Priority	Priority of Agents working for this application.
Application Status	Status of the application corresponding to the value of `application_status_raw`.
Authorization ID	Authorization ID of the user who invoked the application being monitored. On a DDCS gateway node, this is the user's authorization ID on the host.
Average Sort Time per Sort	Average sort time per sort for the statement, which equals: `stmt_sorts/total_sort_time`
Client Database Alias	Alias of the database provided by the application to connect to the database.
Client Node Number	`client_nname` in the database manager configuration file at the client node.
Commit SQL Statements	Number of commit SQL statements.
Coordinator Agent Process Identifier	Process ID (UNIX systems) or thread ID (Windows systems) of the coordinator Agent for the application.
Current Agents Waiting on Locks	Number of applications waiting for a lock held by other applications.
Dynamic SQL Statements	Number of dynamic SQL statements.
Exclusive Lock Escalations	Number of exclusive lock escalations.
Execution Elapsed Time	Sum of the host execution times (in milliseconds) for all the statements that were executed for a particular application.

Table 2–17 (Cont.) Agent Monitoring Metrics

Metric	Description and User Action
Execution Identifier	ID that the user specified when logging in to the operating system. This ID is distinct from the Authorization ID, which the user specifies when connecting to the database.
Failed SQL Statements	Number of failed SQL statements.
Host CPU Usage Per Sec (%)	Ratio of CPU time utilized in the last interval, which equals: `elapsed_exec_time_ms - elapsed_exec_time_ms / __interval` You can set the desired value for the warning and critical thresholds to monitor any adverse conditions.
Lock Timeouts	Number of lock timeouts for the application.
Lock Escalations	Number of lock escalations.
Number of Deadlocks	Total number of deadlocks that have occurred.
Number of Lock Waits	Number of times the application waited for locks.
Number of Locks Currently Held	Number of locks currently held by an application.
Number of Rows Read	Total number of rows read.
Number of Rows Written	Total number of rows written.
Number of Sorts	Number of sorts performed by the statement.
Prefetch Wait Time	IO wait time.
Rows Read	Number of rows read.
Rows Written	Number of rows written.
Rollback SQL Statements	Number of rollback SQL statements.
Start Time	Start time of the statement.
Statement Operation	Statement operation value.
Statement System CPU Time	Total system time used by DBM for processing by the Agent/statement.
Statement User CPU Time	Total user time used by DBM for processing by the Agent/statement.
Statement Text	Whole query or the statement executed.
Statement Type	Number of requests to perform a direct write of one or more sectors of data.
Statement User CPU Time	Total user time DBM used for processing by the Agent/statement.
Static SQL Statements	Number of static SQL statements.
Time Waited on Locks	Total time the application waited for locks.
Total Sorts	Total number of sorts.
Total Sort Time (ms)	Total time used for sorting.
Total CPU Time	Total CPU time, which equals: `stmt_usr_cpu_time + stmt_sys_cpu_time`
Total CPU Time Used by Agent	Total CPU time, which equals: `agent_usr_cpu_time + agent_sys_cpu_time`

2.4.2 Database Monitoring Metrics

The metrics in this category return snapshot information from the database and `detail_log` logical data groups.

Default Collection Interval — Every 15 minutes

Table 2–18 Database Monitoring Metrics

Metric	Description
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Allocated Database Size (Bytes)	Capacity of the database. (Not available in partitioned databases.)
Application Average Lock Wait Time (ms)	Average waiting time for locks, which equals: `lock_wait_time/lock_wait` High wait for an application can mean that the application is degrading performance.
Commit SQL Statements	Number of applications waiting for a lock on an object in the database. Number of commit SQL statements.
Current Agents Waiting on Locks	Number of applications waiting for a lock on an object in the database.
Database Commit SQL Statements Rate	Commit SQL statement reading rate, which equals: `commit_sql_stmts - _commit_sql_stmts/ __interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Database Connection Time	Last time the database was connected.
Database Disconnection Time	Last disconnection time for the database.
Database Dynamic SQL Statements Rate	Dynamic SQL statement reading rate, which equals: `dynamic_sql_stmts - _dynamic_sql_stmts/ __interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Database Failed SQL Statements Rate	Failed SQL statement reading rate, which equals: `failed_sql_stmts - _failed_sql_stmts/ __interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Database Location	Location of the database. Local — if `db_location_raw` = 0 Remote — if `db_location_raw` = 1
Database Location Raw	Location of the database (local or remote).
Database Path	Physical disk location of the database.
Database Rollback SQL Statements Rate	Rollback SQL statement reading rate, which equals: `rollback_sql_stmts - _rollback_sql_stmts/ __interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Database Size	Size of the database in bytes.
Database Space Utilization (%)	Total percentage space utilization in the database, which equals: `(db_size/db_capacity) *100`
Database Static SQL Statements Rate	Static SQL statement reading rate, which equals: `static_sql_stmts - _static_sql_stmts/ __interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Database Status	Status of the database.

Table 2–18 (Cont.) Database Monitoring Metrics

Metric	Description
Deadlocks Rate	Rate of deadlocks, which equals: `(deadlocks-_deadlocks)/_interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Dynamic SQL Statements	Number of dynamic SQL statements.
Exclusive Lock Escalation Rate	Rate of exclusive lock escalations, which equals: `(x_lock_escals-_x_lock_escals)/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Exclusive Lock Escalations	Number of exclusive lock escalations.
Failed SQL Statements	Number of failed SQL statements.
Instance ID	Name of the database manager instance for which the snapshot was taken.
Internal Deadlock Rollback Rate	Rate of internal deadlock rollbacks, which equals: `(int_deadlock_rollbacks-_int_deadlock_rollbacks) /__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Lock Escalations	Number of lock timeouts for the application.
Lock Escalation Rate	Rate of lock escalations, which equals: `(lock_escals-_lock_escals)/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Lock Timeouts	Number of lock escalations.
Locks Timeout Rate	Rate of lock timeouts. The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Number of Current Locks Held	Number of locks currently held by an application.
Number of Currently Active Connections	Total number of active connections in the database.
Number of Currently Connected Applications	Total number of currently connected applications.
Number of Deadlocks	Total number of deadlocks that have occurred.
Number of Internal Rollbacks	Total number of internal deadlock rollbacks.
Number of Lock Waits	Number of times the application waited for locks.
Percentage Applications Waiting on Locks	Ratio of applications waiting on locks, which equals: `(locks_waiting /appls_cur_cons)*100` If a large number of applications are waiting on locks, this indicates a possible performance degradation.
Rollback SQL Statements	Number of rollback SQL statements.
Static SQL Statements	Number of static SQL statements.
Time Waited on Locks	Total time the application waited for locks.
Total Locklist Memory in Use	Total lock list memory in use.

2.4.3 Database Backup Metrics

The metrics in this category provide information regarding the last database backup.

Default Collection Interval — Every 2 hours

Table 2–19 Database Backup Metrics

Metric	Description
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Instance ID	Name of the database manager instance for which the snapshot was taken.
Database Last Backup	Last backup of the database.

2.4.4 General Information Metrics

The metrics in this category provide basic information about the status of the databases and some of the important configuration values.

Default Collection Interval — Every 15 minutes

Table 2–20 General Information Metrics

Metric	Description and User Action
DB2 Start Time	Date and time that the database manager was started using the `db2start` command.
DB Status Value	Status of the DB2 instance.
Agents Registered	Number of Agents registered in the DBM instance that is being monitored.
Communication Private Memory	Amount of communication private memory.
Sort Heap Allocated	Amount of memory allocated to sort heap.
Sort Heap Threshold	Maximum number of private memory pages to be used for private sorts, or the maximum number of shared memory pages to be used for shared sorts. If the sort is a private sort, this parameter affects Agent private memory. If the sort is a shared sort, this parameter affects the database shared memory. Each sort has a separate sort heap that is allocated as needed by the Database Manager. This sort heap is the area where data is sorted. If directed by the optimizer, a smaller sort heap than the one specified by this metric is allocated using information provided by the optimizer.
DB2 Status	Status of the DB2 instance. If DB2 status equals: 0 — Active 1 — Quiesce Pending 2 — Quiesced
Sort Heap Utilization	Sort heap utilization, which equals: `sort_heap_allocated / sortheap_threshold *100` Increase the value of the configuration parameter, `sheapthres`.

2.5 Performance

Performance metrics provide information regarding the performance at various snapshot levels. Performance metrics consist of the following categories:

- Memory Manager
- Sort Heap
- Bufferpool Database Performance
- Bufferpool Performance

- Cache Statistics
- Agent Performance
- Log IO Performance
- Non-Buffered IO Activity

2.5.1 Agent Performance Metrics

The metrics in this category provide performance metrics for the Agent at the application snapshot level.

Default Collection Interval — Every 15 minutes

Table 2–21 Agent Performance Metrics

Metric	Description and User Action
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Cleans for Steals	Number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.
Cleans for Threshold	Number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.

2.5.2 Bufferpool Database Performance Metrics

The metrics in this category provide performance metrics for all the bufferpools in the database.

Default Collection Interval — Every 15 minutes

Table 2–22 Bufferpool Database Performance Metrics

Metric	Description and User Action
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Average Data Write Rate	Pool data write rate, which equals: `(pool_data_writes/pool_write_time)`
Average Index Write Rate	Pool index write rate, which equals: `(pool_index_writes/pool_write_time)`
Average Page Read Rate	The rate equals: `(pool_data_p_reads + pool_index_p_reads )/pool_read_time`
Average Page Write Rate	The rate equals: `(pool_data_writes + pool_index_writes)/pool_write_time`
Database Buffer Pool Hit Ratio (%)	The ratio equals: `(1-((pool_data_p_reads + pool_index_p_reads)/ (pool_index_l_reads + pool_index_l_reads ))*100` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Logical Data Read Rate	The rate equals: `(pool_data_l_reads /pool_read_time)`
Logical Index Read Rate	The rate equals: `(pool_index_l_reads /pool_read_time)`

Table 2–22 (Cont.) Bufferpool Database Performance Metrics

Metric	Description and User Action
Physical Index Read Rate	The rate equals: `(pool_index_p_reads/pool_read_time)`
Physical Index Read Rate	The rate equals: `pool_data_p_reads/pool_read_time`
Pool Logical Data Reads	This count includes accesses to data that is already in the buffer pool when the database manager needs to process the page and read into the buffer pool before the database manager can process the page.
Pool Logical Index Reads	Number of logical read requests to get index pages into the buffer pool.
Pool Physical Data Reads	Number of read requests that required I/O to get data pages into the buffer pool.
Pool Physical Index Reads	Number of physical read requests to get index pages into the buffer pool.
Pool Asynchronous Data Reads	Number of data pages read asynchronously to the buffer pool by prefetchers.
Pool Asynchronous Data Writes	Number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner or a prefetcher.
Pool Asynchronous Index Reads	Number of index pages read asynchronously to the buffer poll by prefetchers.
Pool Asynchronous Index Writes	Number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner or a prefetcher.
Pool Asynchronous Read Time	Number of times a buffer pool data page was physically read from disk by an asynchronous page prefetcher.
Pool Asynchronous Write Time (ms)	Number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner or prefetcher.
Pool Data Writes	Number of times the buffer pool data page was physically written to the disk.
Pool Index Writes	Number of times the buffer pool index page was physically written to the disk.
Pool Logical Data Reads	This count includes accesses to data that is already in the buffer pool when the database manager needs to process the page and read into the buffer pool before the database manager can process the page.
Pool Logical Index Reads	Indicates the number of logical read requests to get index pages into the buffer pool.
Pool Physical Index Reads	Indicates the number of physical read requests to get index pages into the buffer pool.
Pool Physical Data Reads	Number of read requests that required I/O to get data pages into the buffer pool.
Pool Read Time (ms)	Provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
Pool Write Time (ms)	Total amount of time spent physically writing data or index pages from the buffer pool to disk.
Synchronous Data Read Rate	Total synchronous read rate, which equals: `((pool_read_time-pool_async_read_time == 0) ? 0 :((pool_data_p_reads-pool_async_data_reads )/(pool_read_time-pool_async_read_time)))`

Table 2–22 (Cont.) Bufferpool Database Performance Metrics

Metric	Description and User Action
Synchronous Data Write Rate	Pool data synchronous write rate, which equals: `((pool_data_writes - pool_async_data_writes )/(pool_write_time-pool_async_write_time))`
Synchronous Index Read Rate	Index synchronous read rate, which equals: `(pool_data_p_reads-pool_async_index_reads )/(pool_read_time-pool_async_read_time)`
Synchronous Index Write Rate	Index synchronous write rate, which equals: `((pool_index_writes - pool_async_index_writes )/(pool_write_time-pool_async_write_time))`

2.5.3 Bufferpool Performance Metrics

The metrics in this category provide performance metrics for the individual bufferpools in the database.

Default Collection Interval — Every 15 minutes

Table 2–23 Bufferpool Performance Metrics

Metric	Description and User Action
Buffer Pool Name (key column)	Name of the bufferpool.
Database Buffer Pool Hit Ratio (%)	Buffer pool hit ratio, which equals: `(1-((pool_data_p_reads + pool_index_p_reads)/(pool_index_l_reads + pool_index_l_reads ))*100)` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Database Name	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Database Alias	Alias for the database.
Database Path	Physical location of the database.
Pool Asynchronous Data Reads	Number of data pages read asynchronously to the buffer poll by prefetchers.
Pool Asynchronous Data Writes	Number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner or a prefetcher.
Pool Asynchronous Index Reads	Number of index pages read asynchronously to the buffer poll by prefetchers.
Pool Asynchronous Index Writes	Number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner or prefetcher.
Pool Asynchronous Read Time (ms)	Number of times a buffer pool data page was physically read from disk by an asynchronous page prefetcher.
Pool Asynchronous Write Time (ms)	Number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner or prefetcher.
Pool Data Writes	Number of times the buffer pool data page was physically written to the disk.
Pool Index Writes	Number of times the buffer pool index page was physically written to the disk.
Pool Logical Data Reads	This count includes accesses to data that is already in the buffer pool when the database manager needs to process the page and read into the buffer pool before the database manager can process the page.
Pool Logical Index Reads	Indicates the number of logical read requests to put index pages into the buffer pool.

Table 2–23 (Cont.) Bufferpool Performance Metrics

Metric	Description and User Action
Pool Physical Index Reads	Indicates the number of physical read requests to put index pages into the buffer pool.
Pool Physical Data Reads	Number of read requests that required I/O to put data pages into the buffer pool.
Pool Read Time (ms)	Provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from the disk to the buffer pool.
Pool Write Time (ms)	Total amount of time spent physically writing data or index pages from the buffer pool to the disk.

2.5.4 Cache Statistics Metrics

The metrics in this category provide performance information for the package and the catalog cache of the database.

Default Collection Interval — Every 15 minutes

Table 2–24 Cache Statistics Metrics

Metric	Description and User Action
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Catalog Cache Hit Ratio (%)	Catalog cache hit ratio, which equals: `1-(cat_cache_inserts /cat_cache_lookups))*100` The hit ratio is a percentage indicating how well the catalog cache is helping to avoid actual accesses to the catalog on disk. A high ratio indicates it is successful in avoiding actual disk I/O accesses.
Catalog Cache Inserts	Number of inserts performed. The hit ratio is 1-(CCI/CCL).
Catalog Cache Lookups	Number of times the catalog cache was referenced to obtain table description information.
Catalog Cache Overflows	Number of times that the catalog cache overflowed the bounds of its allocated memory.
Package Cache Hit Ratio (%)	Package cache hit ratio, which equals: `1-(pkg_cache_inserts /pkg_cache_lookups))*100` The hit ratio is a percentage indicating how well the package cache is helping to avoid reloading packages and sections for static SQL from the system catalogs as well as helping to avoid recompiling dynamic SQL statements. A high ratio indicates it is successful in avoiding these activities.
Package Cache Inserts	Number of inserts performed. The hit ratio is 1-(PCI/PCL).
Package Cache Lookups	Number of times the package cache was referenced to obtain a section or a package.
Package Cache Max Used (Bytes)	Largest size reached by the package cache.
Package Cache Overflows	Number of times that the package cache overflowed the bounds of its allocated memory.

2.5.5 Log I/O Performance Metrics

The metrics in this category provide performance information for the log input and output including the number of reads and writes in the logs.

Default Collection Interval — Every 15 minutes

Table 2–25 Log I/O Performance Metrics

Metric	Description
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Cleans Logging Threshold	Number of times a page cleaner was invoked because the logging space used had reached a predefined criterion for the database.
Log Reads	Number of log reads.
Log Writes	Number of log writes.

2.5.6 Memory Manager Metrics

The metrics in this category provide the values of the workspace provided and the locklist set in the database.

Default Collection Interval — Every 15 minutes

Table 2–26 Memory Manager Performance Metrics

Metric	Description
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Private Workspace Max Used	Largest size that can be reached by private workspace.
Shared Workspace Max Used	Largest size that can be reached by shared workspace.
Total Locklist Memory in Use	Total amount of lock list memory that is in use.

2.5.7 Sort Heap Metrics

The metrics in this category display the various performance values associated with the sortheap.

Default Collection Interval — Every 15 minutes

Table 2–27 Sort Heap Performance Metrics

Metric	Description and User Action
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Active Sorts	Number of sorts in the database that currently have an allocated sort heap.
Average Active Sorts Rate	Rate of active sorts in the last interval, which equals: `(active_sorts-_active_sorts )/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Hash Join Overflow Rate	Rate of hash join overflow in the last interval, which equals: `(hash_join_overflow-_hash_join_overflow) /__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Hash Join Rate	Rate of hash joins in the last interval, which equals: `(total_hash_joins-_total_hash_joins )/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.

Table 2–27 (Cont.) Sort Heap Performance Metrics

Metric	Description and User Action
Average Hash Join Small Overflow Rate	Rate of the small hash join overflow in the last interval, which equals: `(hash_join_small_overflow-_hash_join_small_overflow )/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Hash Loops Rate	Rate of hash loops in the last interval, which equals: `(total_hash_loops-_total_hash_loops)/ __interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Piped Sorts Rejected Rate	Rate of piped sorts rejection in the last interval, which equals: `((piped_sorts_requested-piped_sorts_accepted)-(_piped_sorts_ requested-_piped_sorts_accepted)) /__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Post Threshold Joins Rate	Rate of post threshold joins in the last interval, which equals: `(post_threshold_ hash_joins-_post_threshold _hash_joins)/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Post Threshold Sorts Rate	Rate of post threshold sorts in the last interval, which equals: `(post_threshold_sorts-_post_threshold_sorts ) /__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Sort Heap Pages Used	Average sort heap space used, which equals: `(sort_heap_allocated /active_sorts)` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Sort Time (ms)	Average time per sort, which equals: `(total_sort_time/total_sorts)` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Sorts Overflow Rate	Rate of sort overflow in the last interval, which equals: (sort_overflow-_sort_overflow)/__interval The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Sorts Rate	Rate of sorts in the last interval, which equals: `(total_sorts-_total_sorts )/__interval` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Piped Sorts Accepted	Number of piped sorts that have been accepted.
Piped Sorts Requested	Number of piped sorts that have been requested.
Post Threshold Hash Sorts	Total number of times that a hash join heap request was limited due to concurrent use of shared or private sort heap space.
Post Threshold Sorts	Number of sorts that have requested heaps after the sort heap threshold has been exceeded.
Sort Overflows	Total number of sorts than ran out of sort heap and may have required disk space for temporary storage.
Sorts Overflow Rate	Sorts overflow ratio, which equals: `((sort_overflow/total_sorts )*100)` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.

Table 2–27 (Cont.) Sort Heap Performance Metrics

Metric	Description and User Action
Total Hash Join Overflows	Number of times that hash join data exceeded the available sort heap space.
Total Hash Join Small Overflow	Number of times that hash join data exceeded the available sort heap space by less than 10%.
Total Hash Joins	Total number of hash joins executed.
Total Hash Loops	Total number of hash loops executed.
Total Sort Heap Allocated	Total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
Total Sorts	Number of sorts that have been executed.
Total Sort Time (ms)	Time spent in sorts.

2.5.8 Non-Buffered I/O Activity Metrics

The metrics in this category display the various performance values related to the non-buffered I/O activities that do not use the buffer pool.

Default Collection Interval — Every 15 minutes

Table 2–28 Non-Buffered I/O Activity Metrics

Metric	Description
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Average Data Read Rate	Direct read rate, which equals: `direct_reads/direct_read_time` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Average Data Write Rate	Direct write rate, which equals: `direct_writes/direct_write_time` The desired value can be set for the warning and critical thresholds to monitor any adverse conditions.
Direct Read Requests	Number of requests to perform a direct read of one of more sectors of data.
Direct Read Time (ms)	Elapsed time in milliseconds required to perform the direct reads.
Direct Reads	Number of read operations that do not use the buffer pool.
Direct Write Requests	Number of requests to perform a direct write of one or more sectors of data.
Direct Write Time (ms)	Elapsed time in milliseconds required to perform the direct writes.
Direct Writes	Number of write operations that do not use the buffer pool.

2.6 Response Metrics

The metrics in this category provide information about the response of the IBM DB2 database in the instance.

Default Collection Interval — Every 5 minutes

Table 2–29 Response Metrics

Metric	Description
Database Name	Name of the database.
DB Status	Status of the database: 0 — Active 1 — Quiesce pending 2 — Quiesced 3 — Roll forward
DB Status Value	Status value of the database.
DB2 Status	Status value of the DB2 instance.
DB2 Status Value	Status of the DB2 instance: 0 — Active 1 — Quiesce pending 2 — Quiesced
Name	Database host name.
Status	Status of the database. The database is up if the status is 0. Otherwise, it is down.

2.7 Storage Information

Storage Information metrics provide information about the storage objects, such as the tablespace and data files. Storage Information metrics consist of the following categories:

- Log Storage
- Tablespace
- Data Files

2.7.1 Data Files Storage Metrics

The metrics in this category provide information about the file properties for the database data storage files.

Default Collection Interval — Every 2 hours

Table 2–30 Tablespace Storage Metrics

Metric	Description
Data File Identifier (key column)	Unique identifier for the data file.
Table Name	Name of the particular table in the database where the file resides.
Table Schema	Schema of the table in which the file resides.
Table Type	Type of table to which the file belongs.
Page Reorganizations	Number of page reorganizations.
Overflow Accesses	Number of overflow accesses.

2.7.2 Log Storage Metrics

The metrics in this category provide information about the log storage properties for the database.

Default Collection Interval — Every 30 minutes

Table 2–31 Log Storage Metrics

Metric	Description and User Action
Database Name (key column)	Real name of the database for which information is collected or to which the application is connected. This is the name the database was given when created.
Allocated Secondary Log Size	Allocated size of the secondary log.
Log Space Utilization	Space utilization of log files.
Total Log Available	Total number of pages available for logging.
Total Log Space Used	Total number of pages used for logging.

2.7.3 Tablespace Storage Metrics

The metrics in this category provide information about the individual tablespace properties for all database tablespaces.

Default Collection Interval — Every 15 minutes

Table 2–32 Tablespace Storage Metrics

Metric	Description
Tablespace Identifier (key column)	Identifier for the tablespace.
Tablespace Name (key column)	Name of the tablespace.
Bufferpool Identifier	Bufferpool identifier for the tablespace.
Extent Size	Extent size for the tablespace.
Number of Containers	Number of containers.
Number of Ranges	Number of ranges.
Page Size	Page size for the tablespace.
Prefetch Size	Prefetch size for the tablespace.
Tablespace Free Pages	Number of pages in a tablespace that will become free if all pending transactions are committed or rolled back, and new space is requested for an object.
Tablespace State	State of the tablespace.
Tablespace Total Pages	Total number of pages in a tablespace.
Tablespace Type	Type of tablespace.
Tablespace Usable Pages	Total number of pages in a tablespace minus overhead pages.
Tablespace Used Pages	Total number of pages currently used (not free) in a tablespace.

2.8 System Configuration Information

System Configuration Information metrics collect the information related to the database software. Each installation of the database software provides an instance to store the data. System Configuration Information metrics consist of the following categories:

- Database System Information
- Product Information
- Partition Information
- Instance Information

- Registry Settings

2.8.1 Database System Information Metrics

The metrics in this category provide information about the various system configuration metrics for the database system, including the name and operating system properties.

- Table Name — MGMT_EMX_IBMDB2_DBSYS
- View Name — MGMT_EMX_IBMDB2_DBSYS_VIEW

Default Collection Interval — Every 24 hours

Table 2–33 Database System Information Metrics

Metric	Description
Server Name	Name of the database server.
OS Name	Name of the operating system.
OS Version	Version of the operating system.
OS Release	Release of the operating system.
Total Number of CPUs	Total number of CPUs for the operating system on which the database is installed.
Total Number of Configured CPUs	Total number of configured CPUs on which the database is installed.
Total Memory (mb)	Amount of memory.

2.8.2 Instance Information Metrics

The metrics in this category return the parameters that provide information about database instances.

- Table Name — MGMT_EMX_IBMDB2_DBINST
- View Name — MGMT_EMX_IBMDB2_DBINST_VIEW

Default Collection Interval — Every 24 hours

Table 2–34 Instance Information Metrics

Metric	Description
Instance Name (key column)	Name of the instance.
Number of DB Partitions	Number of database partitions.
Bit Size of Current Instance	Bit size of the current instance (32 or 64).
Release Number	Internal release number, as returned by the db2level command; 9 for example, 03030106.
Service Level	Service level, as returned by the db2level command; for example, DB2 v8.1.1.80.
Build Level	Build level, as returned by the db2level command; for example, n041021.
Program Temporary Fix	Program temporary fix (PTF) identifier, as returned by the db2level command; for example, U498350.
Fix Pack Number	FixPak number, as returned by the db2level command.

2.8.3 Product Information Metrics

The metrics in this category provide information about the installed IBM DB2 database product.

- Table Name — MGMT_EMX_IBMDB2_DBPRO
- View Name — MGMT_EMX_IBMDB2_DBPRO_VIEW

Default Collection Interval — Every 24 hours

Table 2–35 Product Information Metrics

Metric	Description
Database Path	Installation path of the database.
Product	Installed product.
Version	Version of the installed product.

2.8.4 Partition Information Metrics

The metrics in this category return parameters about parallel operations and partitioned database environments.

- Table Name — MGMT_EMX_IBMDB2_DBPART
- View Name — MGMT_EMX_IBMDB2_DBPART_VIEW

Default Collection Interval — Every 24 hours

Table 2–36 Partition Information Metrics

Metric	Description
Partition Number (key column)	Partition number where the database is installed.
Host Name	Host name of the machine where the db2 database software is installed.
Port Number	TCP/IP port number to communicate with the database.
Switch Name	Name of the switch where the database is connected.

2.8.5 Registry Settings Metrics

The metrics in this category provide information about the various registry parameters for the database.

- Table Name — MGMT_EMX_IBMDB2_DBREGSET
- View Name — MGMT_EMX_IBMDB2_DBREGSET_VIEW

Default Collection Interval — Every 24 hours

Table 2–37 Registry Settings Metrics

Metric	Description
Database Registry Variable (key column)	Name of the DB2 registry variable.
Database Registry Value	Current setting of the DB2 registry variable.
Is Aggregate	Indicates whether or not the DB2 registry variable is an aggregate variable. Possible return values are 0 if it is not an aggregate variable, and 1 if it is an aggregate variable.

Table 2–37 (Cont.) Registry Settings Metrics

Metric	Description
Aggregate Type	Name of the aggregate if the DB2 registry variable is currently getting its value from a configured aggregate. If the registry variable is not being set through an aggregate, or is set through an aggregate but has been overridden, the value of AGGREGATE_ NAME is NULL.
Level Value	Level at which the DB2 registry variable acquires its value. Possible return values and the corresponding levels that they represent are I, G, N, or E.
Level	Values are: Instance — If level1_raw = I Global — If level1_raw = G Database Partition — If level1_raw = N Otherwise, the value is Environment.