Oracle® Enterprise Manager

Framework, Host, and Services Metric Reference Manual

11*g* Release 1 (11.1.0.1)

E17018-01

February 2011

ORACLE®

Oracle Enterprise Manager Framework, Host, and Services Metric Reference Manual 11*g* Release 1 (11.1.0.1)

E17018-01

Contents

2 Host

3 OMS and Repository

5 Web Application Metrics

Preface

This manual is a compilation of the Enterprise Manager framework, host, and services target metrics provided in Oracle Enterprise Manager.

Audience

This document is intended for Oracle Enterprise Manager users interested in Enterprise Manager framework, host, and services target metrics.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Enterprise Manager 10*g* Release 4 documentation set:

- *Oracle Enterprise Manager Grid Control Basic Installation Guide*
- *Oracle Enterprise Manager Grid Control Advanced Installation and Configuration Guide*
- *Oracle Enterprise Manager Concepts*
- *Oracle Enterprise Manager Grid Control Quick Start Guide*
- *Oracle Enterprise Manager Administration*
- *Oracle Enterprise Manager Oracle Database and Database-Related Metric Reference Manual*

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 *Oracle Enterprise Manager Framework, Host, and Services Metric Reference Manual* (hereafter referred to as the *Framework, Host, and Services Metric Reference Manual*) lists all the framework, host, and services target metrics that Enterprise Manager monitors. This manual compiles in one place all the framework, host, and services target metric help available online, eliminating the need to have the Grid Control Console up and running.

This preface describes:

How to Use This Manual

This manual contains a chapter for each Enterprise Manager framework, host, and services target for which there are metrics.

The metrics in each chapter are in alphabetical order according to category.

Metric Information

The information for each metric comprises a description, summary of the metric's "vital statistics", data source (if available), and user action. The following list provides greater detail:

- Description

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

- Metric Summary

  Explains in table format the target version, collection frequency, upload frequency, operator, default warning threshold, default critical threshold, consecutive number of occurrences preceding notification, and alert text for the metric.

- Data Source

  How the metric is calculated. In some metrics, data source information is not available.

- User Action

  Suggestions of how to solve the problem causing the alert.

Examples of Metric Summary Tables

This section provides examples of Metric Summary tables you will see in the *Oracle Enterprise Manager Framework, Host, and Services Metric Reference Manual*.

When default thresholds are not defined for a metric, only the target version and collection frequency are available.

Target Version	Collection Frequency
All Versions	Every 15 Minutes

The following table shows a metric where the server evaluation frequency is the same as the collection frequency.

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 10 Minutes	After Every Sample	>	10000000	12500000	1	Bytes sent by the server are %value%

The following table shows a metric where the server evaluation frequency is different from the collection frequency.

Target Version	Server Evaluation Frequency	Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
10.1.0.x	Every Minute	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	2	Generated By Database Server

Definitions of Columns in Metric Summary Tables

As previously mentioned, the Metric Summary table is part of the overall metric information. The following table provides descriptions of columns in the 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.
Evaluation and Collection Frequency	The rate at which the metric is collected and evaluated to determine whether it has crossed its threshold. The evaluation frequency is the same as the collection frequency.
Server Evaluation Frequency	The rate at which the metric is evaluated to determine whether it has crossed its threshold. For server-generated alerts, the evaluation frequency is determined by Oracle Database internals. For example, if the evaluation frequency is 10 minutes, then when the Average File Write Time degrades to the point an alert should trigger, it could be almost 10 minutes before Enterprise Manager receives indication of the alert. This column is present in the Metric Collection Summary table only for Oracle Database 10*g* metrics.
Collection Frequency	The rate at which the Management Agent collects data. The collection frequency for a metric comes from the Enterprise Manager default collection file for that target type.

Column Header	Column Definition
Upload Frequency	The rate at which the Management Agent moves data to the Management Repository. For example, upload every n^{th} collection. The upload frequency for a metric comes from the Enterprise Manager default collection file for that target type. This column is present in the Metric Collection Summary table only when the Upload Frequency is different from the Collection Frequency.
Comparison Operator	The comparison method Enterprise Manager uses to evaluate the metric value against the threshold values.
Default Warning Threshold	Value that indicates whether a warning alert should be initiated. If the evaluation of the warning threshold value returns a result of TRUE for the specified number of consecutive occurrences defined for the metric, an alert triggers at the warning severity level.
Default Critical Threshold	Value that indicates whether a critical alert should be initiated. If the evaluation of the critical threshold value returns a result of TRUE for the specified number of consecutive occurrences defined for the metric, an alert triggers at the critical severity level.
Consecutive Number of Occurrences Preceding Notification	Consecutive number of times a metric's value reaches either the warning threshold or critical threshold before a notification is sent.
Alert Text	Message indicating why the alert was generated. Words that display between percent signs (%) denote variables. For example, `Disk Utilization for %keyValue% is %value%%` could translate to `Disk Utilization for d0 is 80%`.

Abbreviations and Acronyms

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

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

Background Information on Metrics, Thresholds, and Alerts

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

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

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

- A threshold is reached.
- An alert has been cleared.
- The availability of a monitored service changes. For example, the availability of an application server changes from up to down.

- A specific condition occurs. For example, an alert is triggered whenever an error message is written to a database alert log file.

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

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

Editing

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

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

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

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

Specifying Multiple Thresholds

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

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

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

Accessing Metrics Using the Grid Control Console

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

1. From the Grid Control Console, choose the target.
2. On the target's home page, click **All Metrics** in the Related Links section.
3. On the All Metrics page, choose the metric of interest. If you need help, click **Help**. The help for that metric displays.

Troubleshooting Metrics

In the unlikely situation that a metric does not report a correct value, you need to determine if the problem is related to the:

- Metric providing the wrong values or failing with an error, or

- If the problem is *after* the Management Agent in the execution flow of the metric, that is, the metric value is correct but, for some reason, the data is not reaching the Oracle Management Service.

To aid you in this determination, Oracle provides the Metric Browser; a troubleshooting tool that can be used with Enterprise Manager to see the raw data being collected by the Management Agent.

Accessing the Metric Browser

When enabled, the Metric Browser can be accessed using a web browser, for example, Netscape, Firefox, and Internet Explorer, using a URL of the form:

```
http|https://<agent_hostname>:<agent_port>/emd/browser/main
```

for example

```
http://myServer.myDomain:3872/emd/browser/main
```

> **Note:** You can determine the protocol (http or https), the host name, and the Management Agent port that should be used from the output of the following command (run on the Management Agent host):
>
> ```
> <agent_home>/bin/emctl status agent
> ```

The Management Agent URL, listed in the output to that command, needs only to have *browser* placed between *emd* and *main*.

By default, the Metric Browser is disabled. When the Metric Browser is disabled, you receive the following error:

```
HTTP Error 403 - Forbidden if the metric browser has not been
enabled.
```

How to Enable the Metric Browser and the Management Agent Browser for the Oracle Management Agent

Follow these steps to enable the Metric Browser.

1. The Metric Browser is enabled by setting the enableMetricBrowser property in the Management Agent's emd.properties file. The location of that file depends on the type of Management Agent you are working with:
   - For the Grid Control (central|standalone) Management Agent, the file is:

     ```
     <AGENT_HOME>/sysman/config/emd.properties
     ```

   - For a clustered (RAC) Management Agent install, the file is:

     ```
     <AGENT_HOME>/<hostname>/sysman/config/emd.properties
     ```

   - For the Database Control Management Agent, the file is:

     ```
     <DATABASE_HOME>/<hostname>_
     <SID>/sysman/config/emd.properties
     ```

   - For Application Server Control Management Agent, the file is:

     ```
     <AS_HOME>/sysman/config/emd.properties
     ```

2. Make a backup copy of the emd.properties file.
3. Edit the file and locate the line that reads:

   ```
   #To enable the metric browser, uncomment the following line
   ```

```
#This is a reloadable parameter
#
#enableMetricBrowser=true
```

4. Uncomment the line: #enableMetricBrowser=true, so that it reads:

   ```
   enableMetricBrowser=true
   ```

5. Reload the Management Agent Configuration using the command:

   ```
   <AGENT_HOME>/bin/emctl reload agent
   ```

6. After reloading the Management Agent, the Metric Browser will be enabled and therefore accessible using a browser.

Running the Metric Collection Outside the Management Agent

Running the metric collection outside the Management Agent is specific to each metric and requires a firsthand knowledge of each specific metric. Each metric has its own method of collecting its data and some metrics cannot be run *standalone* because they are calculated from other metrics.

An example of running the metric collection outside the Management Agent is the command line.

1 Agent

The oracle_emd target is a representation of the Oracle Management Agent. The Oracle Management Agent is the Management Agent used by Oracle Enterprise Manager. This target type exposes useful information required to monitor the performance of the Management Agent.

Most of the help topics in this helpset use the term Management Agent to refer to the Oracle Management Agent.

1.1 Agent Process Statistics

The EMD Process Statistics provides information about the performance and resource consumption of the Management Agent process. This metric is collected by default on an interval of 1038 seconds. A value that can be changed in the default collection for the oracle_emd target.

1.1.1 Agent Resident Memory Utilization (KB)

The amount of resident memory used by the agent and all of its child processes in KB.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 1038 Seconds

Data Source

Varies. On UNIX platforms this is derived from the ps command.

User Action

The default warning and critical threshold values for this metric are set higher than what is expected to be necessary in many cases. You may give a lesser value for the warning and critical thresholds based on the number and types of targets that are being monitored by the Management Agent.

1.1.2 Agent Virtual Memory Utilization (KB)

The amount of virtual memory used by the agent and all of its child processes in KB.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 1038 Seconds

Data Source

Varies. On UNIX platforms this is derived from the ps command.

User Action

The default warning and critical threshold values for this metric are set higher than what is expected to be necessary in many cases. You may give a lesser value for the warning and critical thresholds based on the number and types of targets that are being monitored by the Management Agent.

1.1.3 CPU Usage (%)

The CPU Usage metric provides the CPU consumption as a percentage of CPU time at any given moment in time. The number is a summation of the CPU consumption of the Management Agent process and any of its child processes (and their child processes and so on). Child processes are sometimes created by the Management Agent in the course of evaluating a metric or running a job.

Metric Summary

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

Table 1–1 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 1038 Seconds	After Every Sample	>	10	20	4	Agent CPU consumption is %value%%

Data Source

The source for this metric is the UNIX ps command.

User Action

A large CPU consumption will cause the entire system to slow down. The cause could be the Management Agent process itself or any of its child processes. To analyze what is causing the problem, use the Solaris "top" system command and look out for any Perl or Java processes that seem to be consuming excessive CPU (%).

1.1.4 Number Files Open

This metric records the number of files currently opened by the Management Agent process. The file types that constitute this number are: regular files, links, sockets, directories and name pipes.

Metric Summary

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

Table 1–2 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 1038 Seconds	After Every Sample	>	800	900	2	Number files opened by Agent is %value%

Data Source

The source of this information is the UNIX pfiles command. On non-UNIX platforms this will not be collected. On Windows platforms, refer to the File Handles Open metric.

1.1.5 Number Handles Open

This metric records the number of file handles currently opened by the Management Agent process.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 1038 Hours

Data Source

This metric is collected on Windows platforms and is not collected on UNIX platforms. For UNIX, use the "Number Files Open" instead. It is gathered by an agent api.

1.1.6 Number Threads Created

This metric shows the number of threads currently created by the Management Agent process.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 1038 Seconds

Data Source

The source of this metric is the UNIX ps command.

1.1.7 Process ID

The process ID is the process ID of the Management Agent.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 1038 Seconds

Data Source

The source of this is the Perl getppid function.

1.1.8 Resident Memory Utilization (%)

The Resident Memory Utilization is the physical memory usage as a percentage of total memory available.

Metric Summary

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

Table 1–3 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 1038 Seconds	After Every Sample	>	20	30	1	Agent resident memory utilization is %value%%

Data Source

The source of this information is the UNIX ps system command.

1.1.9 Resident Memory Utilization (KB)

This metric represents the amount of physical memory usage by the Management Agent process and all of its child processes in KB.

Metric Summary

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

Table 1–4 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 1038 Seconds	After Every Sample	>	128000	256000	1	Agent resident memory utilization in KB is %value%

Data Source

The source of this data is the UNIX ps system command.

User Action

The default warning and critical threshold values for this metric are set higher than what is expected to be necessary in many cases. You will probably want to lower the warning and critical thresholds to values that work well for the number and types of targets that are being monitored by the Management Agent.

1.1.10 Virtual Memory Utilization (KB)

The Virtual Memory Utilization (VMU) metric provides a sum of the VMU usage of the Management Agent and all of its child processes (and their child processes and so on). Child processes are sometimes created by the Management Agent in the course of evaluating a metric or running a job.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 1038 Seconds

Data Source

The source of this information is the UNIX ps system command.

User Action

Large virtual memory utilization will also slow the system down. On UNIX machines, use the "top" command to see what processes are consuming this memory. Look out for Perl and Java processes as well as the obvious emdaemon process (the Management Agent process itself.)

1.1.11 Virtual Memory Utilization Growth (%)

Virtual memory utilization growth (%) shows the *percentage* growth of the virtual memory percentage usage of the Management Agent process. For example: if at time t1 (t1 < t2) the usage was a% and at time t2 it was b%, the growth % would be ((b-1)/a)%.

Metric Summary

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

Table 1–5 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 1038 Seconds	After Every Sample	>	.5	2	3	Agent Virtual Memory Growth is %value%%

Data Source

The source of the raw information is the UNIX ps command. From this, we calculate an average over four interval periods and use this as our comparison percentage (that is, how much has the virtual memory usage grown as a percentage of this average.

1.2 Response

The Response metric reports on the availability of the Management Agent.

1.2.1 Status

This metric has a value of 1 if the Management Agent is up and running.

Metric Summary

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

Table 1–6 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	On startup	After Every Sample	=	Not Defined	0	1	Not Defined

Data Source

Not applicable.

User Action

If the value of this metric is not 1, the Management Agent is down and contact with the Management Agent will not exist. In such situations, the Management Agent may need to be restarted.

1.3 Targets not uploading

This category of metrics provides information on the targets that do not upload data.

1.4 Upload Statistics

The Upload Statistics metrics present information on the state of the upload manager and its performance.

1.4.1 Count of targets not uploading data

This metric provides a count of the targets that are not uploading data.

Data Source

The mgmt_targets, mgmt_current_availability tables in the Management Repository.

User Action

Verify the connection between the agent and OMS to which the agent is uploading is working properly. Check for frequent agent restarts, sufficient disk space for the agent upload directory, any severe agent problems logged in agent error logs, severe problems logged in the OMS error log, loader errors logged in the System Errors page.

1.4.2 Number of Files to Upload

This metric shows the number of XML files that are in the $ORACLE_HOME/sysman/emd/upload directory waiting to be uploaded to the repository.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 62 Minutes

Data Source

The source of this information is the Management Agent itself.

User Action

A large number of files in this directory probably indicates that there is a problem uploading files to the repository. Check the emd.trc file for upload errors and act appropriately. The cause may also be a bad network or problems on the repository end.

1.4.3 Size of Files to Upload (MB)

The Size of Files to Upload metric presents the sum of the sizes of all XML files in the upload directory of the Management Agent.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 62 Minutes

Data Source

The source of this information is the Management Agent itself.

User Action

If this metric is large, check the upload directory. If this directory has very few files, it may be they are large. If it has many files, there may be a problem uploading data to

the repository. This may be due to a bad network, bad repository or Management Agent. Check the emd.trc file in the log directory for upload error messages.

1.4.4 Upload Rate (KB/sec)

The upload rate is the average rate in KB/sec at which data is uploaded to the repository.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 62 Minutes

Data Source

The source of this information is the Management Agent itself.

User Action

If the rate is zero or close to zero, there may be problems uploading data or collecting data (because if collections stop for some reason, we have nothing to upload). Check the log files for collection and upload messages.

1.5 User Identification

These metrics provide information about the user running the Management Agent.

1.5.1 Group Name

The name of the group the Management Agent is running under.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this metric is the UNIX id command.

1.5.2 Location

The Location metric shows the directory home of the Management Agent.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this information is the ORACLE_HOME environment variable.

1.5.3 Other Groups

This metric lists the other groups the Management Agent user belongs to.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this metric is the UNIX "id" command.

1.5.4 User Name

The User Name metric provides information on the user that started the Management Agent process.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this data is the UNIX id command.

1.6 User Limit Info

The metrics in the User Limit Info category provide information about the system resources available to the Management Agent.

1.6.1 CoreDump (blocks)

The CoreDump metric shows the maximum size of a core dump file in 512 Kbytes blocks. A value of unlimited means that the only limit is the file system limit.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this information is the UNIX ulimit command.

User Action

This metric shows the maximum size (in 512 Kbyte blocks) of a core dump file. To decrease or increase this limit, use the UNIX ulimit system command.

1.6.2 Data (kbytes)

This metric shows the maximum size of the Management Agent's heap in Kbytes.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source for this information is the UNIX ulimit system command.

User Action

This metric shows the maximum heap size (in kbytes) made available to the Management Agent. To decrease or increase this limit, use the UNIX ulimit system command.

1.6.3 File (blocks)

The File metric lets you know the size of the largest single file allowed by the system the Management Agent is running on. The unit is 512 Kbyte blocks. A value of "unlimited" means that the limit is the file system limit.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source for this information is the UNIX ulimit system command.

User Action

This metric shows the maximum file size (in blocks) allowed by the system that the Management Agent is running on. To decrease or increase this limit, use the UNIX ulimit system command.

1.6.4 NoFiles (descriptors)

The NoFiles metric shows the maximum number of file descriptors that the process can have.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this information is the UNIX system call ulimit.

User Action

If this limit is small (compared to the operating system maximum), it can be changed for the Management Agent process.

1.6.5 Stack (kbytes)

This metric displays the maximum size of the Management Agent's stack in Kbytes.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source for this information is the UNIX ulimit command.

User Action

This metric shows the maximum size (in kbytes) of the Management Agent's stack. To decrease or increase this limit, use the UNIX ulimit system command.

1.6.6 Time (seconds)

The time metric represents, in seconds, the maximum CPU seconds made available to the Management Agent process by the system it is running on. A value of "unlimited" means that the CPU time available to the Management Agent is unrestricted.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

This information is obtained using the UNIX ulimit system command.

User Action

This metric shows the maximum CPU time (in seconds) made available to the Management Agent. To decrease or increase this limit, use the UNIX ulimit system command.

1.6.7 Virtual Mem (kbytes)

The Virtual Mem metric shows the maximum virtual memory size that can be occupied by the Management Agent process. If this value is "unlimited" then the only limit is the operating system limit.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 168 Hours

Data Source

The source of this information is the UNIX ulimit system command.

User Action

If the value of the Virtual Mem metric is too high or too low, you can change the restriction on virtual memory use by the Management Agent process using the limit UNIX command.

2

Host

The host metrics provide description, collection statistics, data source, multiple thresholds (where applicable), and user action information for each metric.

2.1 Aggregate Resource Usage Statistics (By Project)

This metric provides data on aggregate resource usage on a per project basis.

This metric is available only on Solaris version 9 and later.

The following table lists the metrics and their descriptions.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The data source for these metrics is Solaris CIM Object Manager.

Table 2–1 Aggregate Resource Usage Statistics (By Project)

Metric	Description
Cumulative CPU Wait Time (Seconds)	Cumulative number of seconds that this process has spent Waiting for CPU over its lifetime
Cumulative Data Page Fault Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in Data Page Faults over its lifetime
Cumulative Major Page Faults	Cumulative number of Major Page Faults engendered by the process over its lifetime
Cumulative Minor Page Faults	Cumulative number of Minor Page Faults engendered by the process over its lifetime
Cumulative Number Character IO (bytes) Read and Written	Cumulative number of character I/O bytes Read and Written by the process over its lifetime
Cumulative Number of Blocks Read	Cumulative number of blocks Read by the process over its lifetime
Cumulative Number of Blocks Written	Cumulative number of blocks Written by the process over its lifetime
Cumulative Number of Involuntary Context Switches	Cumulative number of Involuntary Context Switches made by the process over its lifetime
Cumulative Number of Messages Received	Cumulative number of Messages Received by the process over its lifetime

Table 2–1 (Cont.) Aggregate Resource Usage Statistics (By Project)

Metric	Description
Cumulative Number of Messages Sent	Cumulative number of Messages Sent by the process over its lifetime
Cumulative Number of Signals Received	Cumulative number of Signals taken by the process over its lifetime
Cumulative Number of System Calls Made	Cumulative number of system calls made by the process over its lifetime
Cumulative Number of Voluntary Context Switches	Cumulative number of Voluntary Context Switches made by the process over its lifetime
Cumulative Project Lock-Wait Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping on User Lock Waits over its lifetime
Cumulative Project Other Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in all other ways over its lifetime
Cumulative Stop Time (Seconds)	Cumulative number of seconds that this process has spent Stopped over its lifetime
Cumulative Swap Operations	Cumulative number of swap operations engendered by the process over its lifetime
Cumulative System Mode Time (Seconds)	Cumulative number of seconds that this process has spent in System mode over its lifetime
Cumulative System Page Fault Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in System Page Faults over its lifetime
Cumulative System Trap Time (Seconds)	Cumulative number of seconds that this process has spent in System Traps over its lifetime
Cumulative Text Page Fault Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in Text Page Faults over its lifetime
Cumulative User Mode Time (Seconds)	Cumulative number of seconds that this process has spent in User mode over its lifetime
Number of Processes Owned by Project	Number of processes owned by the project measured in the aggregate
Project CPU Time (%)	Percent CPU time used by the process
Project Process Memory Size (%)	Ratio of the process resident set size to physical memory
Project's Total Process Heap Size (KiloBytes)	Total number of KiloBytes of memory consumed by the process heap at the time that it is sampled
Project's Total Process Resident Set Size (KiloBytes)	Resident set size of the process in kilobyte
Project's Total Process Virtual Memory Size (KiloBytes)	Resident set size of the process in kilobyte
Total Number of Threads in Project's Processes	Number of threads active in the current Process

2.2 Aggregate Resource Usage Statistics (By User)

This metric provides data on aggregate resource usage on a per user basis.

This metric is available only on Solaris version 9 and later.

The following table lists the metrics and their descriptions.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The data source for these metrics is Solaris CIM Object Manager.

Table 2–2 Aggregate Resource Usage Statistics (By User)

Metric	Description
Cumulative CPU Wait Time (Seconds)	Cumulative number of seconds that this process has spent Waiting for CPU over its lifetime
Cumulative Data Page Fault Sleep Time (Seconds)	Cumulative number of seconds that this process has spent Waiting for CPU over its lifetime
Cumulative Major Page Faults	Cumulative number of Major Page Faults engendered by the process over its lifetime
Cumulative Minor Page Faults	Cumulative number of Minor Page Faults engendered by the process over its lifetime
Cumulative Number Character IO (Bytes) Read and Written	Cumulative number of character I/O bytes Read and Written by the process over its lifetime
Cumulative Number of Blocks Read	Cumulative number of blocks Read by the process over its lifetime
Cumulative Number of Blocks Written	Cumulative number of blocks Written by the process over its lifetime
Cumulative Number of Involuntary Context Switches	Cumulative number of Involuntary Context Switches made by the process over its lifetime
Cumulative Number of Messages Received	Cumulative number of Messages Received by the process over its lifetime
Cumulative Number of Messages Sent	Cumulative number of Messages Sent by the process over its lifetime
Cumulative Number of Signals Received	Cumulative number of Signals taken by the process over its lifetime
Cumulative Number of System Calls Made	Cumulative number of system calls made by the process over its lifetime
Cumulative Number of Voluntary Context Switches	Cumulative number of Voluntary Context Switches made by the process over its lifetime
Cumulative Stop Time (Seconds)	Cumulative number of seconds that this process has spent Stopped over its lifetime
Cumulative Swap Operations	Cumulative number of Swap Operations engendered by the process over its lifetime
Cumulative System Mode Time (Seconds)	Cumulative number of seconds that this process has spent in System mode over its lifetime
Cumulative System Page Fault Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in System Page Faults over its lifetime
Cumulative System Trap Time (Seconds)	Cumulative number of seconds that this process has spent in System Traps over its lifetime

Table 2–2 (Cont.) Aggregate Resource Usage Statistics (By User)

Metric	Description
Cumulative Text Page Fault Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in Text Page Faults over its lifetime
Cumulative User Lock-Wait Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping on User Lock Waits over its lifetime
Cumulative User Mode Time (Seconds)	Cumulative number of seconds that this process has spent in User mode over its lifetime
Cumulative User Other Sleep Time (Seconds)	Cumulative number of seconds that this process has spent sleeping in all other ways over its lifetime
Number of Processes Owned by User	Number of processes owned by the user measured in the aggregate
Total Number of Threads in User's Processes	Number of processes owned by the user measured in the aggregate
User CPU Time (%)	Percent CPU time used by the process
User Process Memory Size (%)	Ratio of the process resident set size to physical memory
User's Total Process Heap Size (KiloBytes)	Total number of kilobytes of memory consumed by the process heap at the time that it is sampled
User's Total Process Resident Set Size (KiloBytes)	Resident set size of the process in kilobytes
User's Total Process Virtual Memory Size (KiloBytes)	Size of the process virtual address space in kilobytes

2.3 Buffer Activity

The Buffer Activity metric provides information about OS memory buffer usage. This metric reports buffer activity for transfers, accesses, and cache (kernel block buffer cache) hit ratios per second.

The data sources for this metric category include the following:

Host	Data Source
Solaris	sar command
HP	sar command
Linux	not available
HP Tru64	table() system call
IBM AIX	sar command
Windows	not available

The following table lists the metrics and their descriptions.

Table 2–3 Buffer Activity Metrics

Metric	Description
Buffer Cache Read Hit Ratio (%)	Number of reads from block devices to buffer cache as a percentage of all buffer reads

Table 2–3 (Cont.) Buffer Activity Metrics

Metric	Description
Buffer Cache Reads (per second)	Number of reads performed on the buffer cache per second. **Note:** This metric is not available on HP Tru64.
Buffer Cache Write Hit Ratio (%)	Number of writes from block devices to buffer cache as a percentage of all buffer writes
Buffer Cache Writes (per second)	Number of writes performed on the buffer cache per second. **Note:** This metric is not available on HP Tru64.
Physical I/O Reads (per second)	Number of reads per second from character devices using physical I/O mechanisms
Physical I/O Writes (per second)	Number of writes per second from character devices using physical I/O mechanisms
Physical Reads (per second)	Number of reads performed per second from block devices to the system buffer cache
Physical Writes (per second)	Number of physical writes from block devices to the system buffer cache

2.4 CPU Usage

The CPU Usage metric provides information about the percentage of time the CPU was in various states, for example, idle state and wait state. The metric also provides information about the percentage of CPU time spent in user and system mode. All data is per-CPU in a multi-CPU system.

On HP Tru64, this information is available as the cumulative total for all the CPUs and not for each CPU which is monitored in the Load metric. Hence, this metric is not available on HP Tru64.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The data sources for this metric category include the following:

Host	Data Source
Solaris	kernel statistics (class cpu_stat)
HP	pstat_getprocessor() system call
Linux	/proc/stat
HP Tru64	not available
IBM AIX	oracle_kstat() system call
Windows	performance data counters

The following table lists the metrics and their descriptions.

Table 2–4 CPU Usage Metrics

Metric	Description
CPU Idle Time (%)	Represents the percentage of time that the CPU was idle and the system did not have an outstanding disk I/O request. This metric checks the percentage of processor time in idle mode for the CPU(s) specified by the Host CPU parameter, such as cpu_stat0, CPU0, or * (for all CPUs on the system).
CPU Interrupt Time (%)	See Section 2.4.1, "CPU Interrupt Time (%)" **Note:** This metric is available only on Windows.
CPU System Time (%)	Represents the percentage of time that the CPU is running in system mode (kernel). This metric checks the percentage of processor time in system mode for the CPU(s) specified by the Host CPU parameter, such as cpu_stat0, CPU0, or * (for all CPUs on the system).
CPU User Time (%)	Represents the portion of processor time running in user mode. This metric checks the percentage of processor time in user mode for the CPU(s) specified by the Host CPU parameter, such as cpu_stat0, CPU0, or * (for all CPUs on the system).
CPU Wait Time (%)	Represents the percentage of time that the CPU was idle during which the system had an outstanding disk I/O request. This metric checks the percentage of processor time in wait mode for the CPU(s) specified by the Host CPU parameter, such as cpu_stat0, CPU0, or * (for all CPUs on the system). **Note:** This metric is not available on Solaris and HP Tru64.

2.4.1 CPU Interrupt Time (%)

Represents the percentage of time that the CPU receives and services hardware interruptions during representative intervals. This metric checks the percentage of processor time in interrupt mode for the CPU(s) specified by the Host CPU parameter, such as cpu_stat0, CPU0, or * (for all CPUs on the system).

This metric is available only on Windows.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "CPU Number" object.

If warning or critical threshold values are currently set for any "CPU Number" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "CPU Number" object, use the Edit Thresholds page. See the Editing Thresholds topic in the Enterprise Manager online help for information on accessing the Edit Thresholds page.

Data Source

The data sources for this metric are Performance Data counters.

2.5 CRS Alert Log

This metric collects certain Cluster Ready Services (CRS) error messages and issues either WARNING or CRITICAL alerts based on the error codes.

2.5.1 Alert Log Name

Shows the name and full path of the Cluster Ready Services (CRS) alert log.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 5 Minutes

2.5.2 Clusterware Service Alert Log Error

Collects CRS-1012, CRS-1201, CRS-1202 and CRS-1401, CRS-1402, CRS-1602 and CRS-1603 messages in the Cluster Ready Services (CRS) alert log at the host level.

CRS-1201, CRS-1401, CRS-1012 alert log messages trigger warning alerts.

CRS-1202, CRS-1402, CRS-1602 and CRS-1603 alert log messages trigger critical alerts.

Metric Summary

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

Table 2–5 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	MATCH	CRS-(1201|1401|1012)	CRS-(1202|1402|1602|1603)	1*	%clusterwareErrStack% See %alertLogName% for details.

* Once an alert is triggered for this metric, it must be manually cleared.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Time/Line Number" object.

If warning or critical threshold values are currently set for any "Time/Line Number" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Time/Line Number" object, use the Edit Thresholds page.

2.5.3 CRS Resource Alert Log Error

Collects CRS-1203, CRS-1205 and CRS-1206 messages in the Cluster Ready Services (CRS) alert log at the host level and issues 'CRS Resource Alert Log Error' alerts at critical level.

Metric Summary

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

Table 2–6 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	MATCH	Not Defined	CRS-120(3|5|6)	1*	%resourceErrStack% See %alertLogName% for details.

* Once an alert is triggered for this metric, it must be manually cleared.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Time/Line Number" object.

If warning or critical threshold values are currently set for any "Time/Line Number" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Time/Line Number" object, use the Edit Thresholds page.

2.5.4 OCR Alert Log Error

Collects CRS-1009 messages in the Cluster Ready Services (CRS) alert log at the host level and issues 'OCR Alert Log Error' type alerts. OCR refers to Oracle Cluster Registry.

Metric Summary

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

Table 2–7 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	MATCH	Not Defined	CRS-1009	1*	%ocrErrStack% See %alertLogName% for details.

* Once an alert is triggered for this metric, it must be manually cleared.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Time/Line Number" object.

If warning or critical threshold values are currently set for any "Time/Line Number" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Time/Line Number" object, use the Edit Thresholds page.

2.6 CRS Nodeapp Status

This metric monitors the status of the following: Node Applications (nodeapps), Virtual Internet Protocol (IP), Global Services Daemon (GSD), and Oracle Notification System (ONS).

2.6.1 Nodeapp Status

Monitors the status of the following: Node Applications (nodeapps), Virtual Internet Protocol (IP), Global Services Daemon (GSD), and Oracle Notification System (ONS). A critical alert is raised for the nodeapp if its status is 'OFFLINE NOT RESTARTING'. A warning alert is raised for the nodeapp if its status is either 'UNKNOWN or OFFLINE'.

Metric Summary

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

Table 2–8 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	MATCH	UNKNOWN | OFFLINE	OFFLINE NOT RESTARTING	1	CRS resource %nodeapps% is %status%

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Nodeapp" object.

If warning or critical threshold values are currently set for any "Nodeapp" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Nodeapp" object, use the Edit Thresholds page.

User Action

Refer to the *Real Application Clusters Administration and Deployment Guide* for Node Applications startup and troubleshooting information.

2.7 CRS Virtual IP Relocation Status

This metric monitors whether there is a Virtual Internet Protocol (IP) relocation taking place. When a Virtual IP is relocated from the host (node) on which it was originally configured, a critical alert is generated.

2.7.1 Current Node

Shows the current host (node) on which the Virtual Internet Protocol (IP) is configured.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 5 Minutes

2.7.2 Virtual IP Relocated

Shows whether the Virtual Internet Protocol (IP) has relocated from the host (node) where it was originally configured. The value is TRUE if relocation happened. Otherwise it is FALSE. When the value is TRUE, a critical alert is raised.

Metric Summary

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

Table 2–9 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	=	Not Defined	TRUE	1	CRS resource %vip% was relocated to %current_node%

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Virtual IP Name" object.

If warning or critical threshold values are currently set for any "Virtual IP Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Virtual IP Name" object, use the Edit Thresholds page.

2.8 Disk Activity

The Disk Activity metric monitors the hard disk activity on the target being monitored. For each device on the system, this metric provides information about access to the device. This information includes: device name, disk utilization, write statistics, and read statistics for the device.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The data sources for this metric category include the following:

Host	Data Source
Solaris	kernel statistics (class kstat_io)
HP	pstat_getdisk system call
Linux	iostat command
HP Tru64	table() system call

Host	Data Source
IBM AIX	oracle_kstat() system call
Windows	performance data counters

The following table lists the metrics and their descriptions.

Table 2–10 Disk Activity Metrics

Metric	Description
Average Disk I/O Service Time (ms)	See Section 2.8.1, "Average Disk I/O Service Time (ms)"
Average Disk I/O Wait Time (ms)	See Section 2.8.2, "Average Disk I/O Wait Time (ms)". **Note:** This metric is not available on Linux.
Average Outstanding Disk I/O Requests	Represents the average number of commands waiting for service (queue length). **Note:** This metric is not available on Linux.
Average Run Time (ms)	Represents the average time spent by the command on the active queue waiting for its execution to be completed. **Note:** This metric is not available on Linux.
Disk Block Writes (per second)	Represents the number of blocks (512 bytes) written per second. **Note:** This metric is not available on HP.
Disk Block Reads (per second)	Represents the number of blocks (512 bytes) read per second. **Note:** On HPUNIX, this metric is named Disk Blocks Transferred (per second).
Disk Device Busy (%)	See Section 2.8.3, "Disk Device Busy (%)". **Note:** On HPUNIX, this metric is named Device Busy (%).
Disk Reads (per second)	Represents the disk reads per second for the specified disk device. **Note:** This metric is not available on HP.
Disk Writes (per second)	Represents the disk writes per second for the specified disk device. **Note:** This metric is not available on HP.

2.8.1 Average Disk I/O Service Time (ms)

Represents the sum of average wait time and average run time.

Metric Summary

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

Table 2–11 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	>	Not Defined	Not Defined	6	Average service time for disk %keyvalue% is %value% ms, crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Disk Device" object.

If warning or critical threshold values are currently set for any "Disk Device" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Disk Device" object, use the Edit Thresholds page.

User Action

This number should be low. A high number can indicate a disk that is slow due to excessive load or hardware issues. See also the CPU in IO-Wait (%) metric.

2.8.2 Average Disk I/O Wait Time (ms)

Represents the average time spent by the command waiting on the queue for getting executed.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Disk Device" object.

If warning or critical threshold values are currently set for any "Disk Device" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Disk Device" object, use the Edit Thresholds page.

User Action

A high figure indicates a slow disk. Use the OS iostat -xn command to check wait time and service time for local disks and NFS mounted file systems. See also the CPU in IO-Wait (%) metric.

2.8.3 Disk Device Busy (%)

Represents the amount of disk space utilization as a percentage of capacity.

Note: On HPUNIX, this metric is named Device Busy (%).

Metric Summary

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

Table 2–12 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	>	80	95	6	Disk Device %keyValue% is %value%% busy.

2.9 Disk Device Errors

The Disk Device Errors metric provides the number of errors on the disk device.

These metrics are available only on Solaris.

Note: For all target versions, the collection frequency for each metric is every 72 hours.

The data source for these metrics is Solaris iostat -e command.

Table 2–13 Disk Device Errors Metrics

Metric	Description
Hard Errors	Represents the error count of hard errors encountered while accessing the disk. Hard errors are considered serious and may be traced to misconfigured or bad disk devices.
Soft Errors	Represents the error count of soft errors encountered while accessing the disk. Soft errors are synonymous to warnings.
Total	Represents the sum of all errors on the particular device.
Transport Errors	Represents the error count of network errors encountered. This generally indicates a problem with the network layer

2.10 Fans

The Fans metric monitors the status of various fans present in the system.

This metric is available only on Dell Poweredge Linux Systems.

2.10.1 Fan Status

Represents the status of the fan.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–14 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of Fan at device %FanIndex% in chassis %ChassisIndex% is %value%, crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis Index" and "Fan Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis Index" and "Fan Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis Index" and "Fan Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: coolingDeviceStatus (1.3.6.1.4.1.674.10892.1.700.12.1.5)

2.10.2 Location

Provides a description of the location of the fan. Example values are "CPU Fan", "PCI Fan", and "Memory Fan".

This metric is available only on Dell Poweredge Linux Systems.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 15 Minutes

Data Source

SNMP MIB object: coolingDeviceLocationName (1.3.6.1.4.1.674.10892.1.700.12.1.8)

2.11 File Access System Calls

The File Access System Calls metric provides information about the usage of file access system calls.

This metric is available on Solaris, HP, and IBM AIX.

2.11.1 Blocks Read by Directory Search Routine (per second)

Represents the number of file system blocks read per second performing direct lookup.

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	sar command
HP	sar command
IBM AIX	sar command

The OS sar command is used to sample cumulative activity counters maintained by the OS. The data is obtained by sampling system counters once in a five-second interval. The results are essentially the number of lookuppn() calls made over this five-second period divided by five.

2.11.2 iget() Calls (per second)

Represents the number of system iget() calls made per second. iget is a file access system routine.

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	kernel memory structure (class cpu_vminfo
HP	sar command
IBM AIX	kernel memory structure (class cpu_vminfo

User Action

This data is obtained using the OS sar command, which is used to sample cumulative activity counters maintained by the OS. The data is obtained by sampling system counters once in a five-second interval. The results are essentially the number of iget() calls made over this five-second period divided by five.

2.11.3 lookuppn() Calls (per second)

Represents the number of file system lookuppn() (pathname translation) calls made per second.

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	sar command
HP	sar command
IBM AIX	sar command

The OS sar command is used to sample cumulative activity counters maintained by the OS. The data is obtained by sampling system counters once in a five-second

interval. The results are essentially the number of lookuppn() calls made over this five-second period divided by five.

2.12 File and Directory Monitoring

The File and Directory Monitoring metric monitors various attributes of specific files and directories. Setting of key value specific thresholds triggers the monitoring of files or directories referred to in the given key value. The operator must specify key value specific thresholds to monitor any file or directory.

The data sources for this metric include the following:

Host	Data Source
Solaris	perl stat command for files; df for directories that are file system mount points; du for directories that are not file system mount points
HP	perl stat command for files; df for directories that are file system mount points; du for directories that are not file system mount points
Linux	perl stat command for files; df for directories that are file system mount points; du for directories that are not file system mount points
HP Tru64	not available
IBM AIX	perl stat command for files; df for directories that are file system mount points; du for directories that are not file system mount points
Windows	not available

2.12.1 File or Directory Attribute Not Found

Reports issues encountered in fetching the attributes of the file or directory. Errors encountered in monitoring the files and directories specified by the key value based thresholds are reported.

Note: This metric is not available on IBM AIX.

Metric Summary

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

Table 2–15 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	!=	Not Defined	0	1	%file_attribute_not_found% .

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "File or Directory Name" object.

If warning or critical threshold values are currently set for any "File or Directory Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "File or Directory Name" object, use the Edit Thresholds page.

2.12.2 File or Directory Permissions

Fetches the octal value of file permissions on the different variations of UNIX operating systems including Linux. Setting a key value specific warning or critical threshold value against this metric would result in the monitoring of a critical file or directory. For example, to monitor the file permissions for file name /etc/passwd, you should set a threshold for /etc/passwd.

Metric Summary

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

Table 2–16 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	!=	Not Defined	Not Defined	1	Current permissions for %file_name% are %file_permissions%, different from warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "File or Directory Name" object.

If warning or critical threshold values are currently set for any "File or Directory Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "File or Directory Name" object, use the Edit Thresholds page.

2.12.3 File or Directory Size (MB)

Fetches the current size of the given file or directory in megabytes. Setting a key value specific warning or critical threshold value against this metric would result in monitoring of a critical file or directory. For example, to monitor the file permissions for directory /absolute_directory_path, you should set a threshold for /absolute_directory_path.

Metric Summary

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

Table 2–17 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	>	Not Defined	Not Defined	1	Size of %file_name% is %file_size% MB, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "File or Directory Name" object.

If warning or critical threshold values are currently set for any "File or Directory Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "File or Directory Name" object, use the Edit Thresholds page.

2.12.4 File or Directory Size Change Rate (KB/minute)

Provides the value for the rate at which the files size is changing. Setting a key value specific warning or critical threshold value against this metric would result in monitoring of the critical file or directory. For example, to monitor the file change rate for the file name /absolute_file_path, the operator should set a threshold for /absolute_file_path.

Metric Summary

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

Table 2–18 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	>	Not Defined	Not Defined	1	%file_name% is growing at the rate of %file_sizechangerate% (KB/hour), crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "File or Directory Name" object.

If warning or critical threshold values are currently set for any "File or Directory Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "File or Directory Name" object, use the Edit Thresholds page.

2.13 Filesystems

The Filesystems metrics provide information about local file systems on the computer.

2.13.1 Filesystem

Represents the name of the disk device resource.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 15 Minutes

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	/etc/mnttab file entries
HP	bdf command
Linux	df command
HP Tru64	df command
IBM AIX	/etc/mnttab file entries
Windows	not available

2.13.2 Filesystem Size (MB)

Represents the total space (in megabytes) allocated in the file system.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 15 Minutes

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	vminfo system
HP	bdf command
Linux	df command
HP Tru64	df command
IBM AIX	stavfs() system call
Windows	not available

2.13.3 Filesystem Space Available (%)

Represents the percentage of free space available in the file system.

Metric Summary

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

Table 2–19 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every 24 Samples	<	20	5	1	Filesystem %keyValue% has %value%% available space, fallen below warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Mount Point" object.

If warning or critical threshold values are currently set for any "Mount Point" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Mount Point" object, use the Edit Thresholds page.

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	stavfs() system call
HP	bdf command
Linux	df command
HP Tru64	df command
IBM AIX	stavfs() system call
Windows	Windows API

User Action

Use the OS du -k command to check which directories are taking up the most space (du -k | sort -rn).

2.13.4 Filesystem Utilization (MB)

Represents the total space, expressed in megabytes, allocated in the file system.

This metric is available only on Windows.

Data Source

The data source for this metric is GetDiskFreeSpaceEx.

2.14 Inventory

The Inventory metric is used for periodic collection of host configuration information. By default, host configuration is collected every 24 hours.

2.15 Kernel Memory

The Kernel Memory metric provides information on kernel memory allocation (KMA) activities.

This metric is available only on Solaris. The data source is the `sar` command. The data is obtained by sampling system counters once in a five-second interval.

The following table lists the metrics and their descriptions.

Table 2–20 Kernel Memory Metrics

Metric	Description
Failed Requests for Large Kernel Memory	Number of requests for large memory that failed, that is, requests that were not satisfied
Failed Requests for Oversize Kernel Memory	Number of oversized requests made that could not be satisfied. Oversized memory requests are allocated dynamically so there is no pool for such requests
Failed Requests for Small Kernel Memory	Number of requests for small memory that failed, that is, requests that were not satisfied
KMA Available for Large Memory Requests (Bytes)	Amount of memory, in bytes, the kernel memory allocation (KMA) has for the large pool; the pool used for allocating and reserving large memory requests.
KMA for Oversize Memory Requests (Bytes)	Amount of memory allocated for oversized memory requests
KMA for Small Memory Requests	Amount of memory, in bytes, the Kernel Memory Allocation has for the small pool; the pool used for allocating and reserving small memory requests
Memory Allocated for Large Memory Requests (Bytes)	Amount of memory, in bytes, the kernel allocated to satisfy large memory requests
Memory Allocated for Small Memory Requests (Bytes	Amount of memory, in bytes, the kernel allocated to satisfy small memory requests

2.16 Load

The Load metric provides information about the number of runnable processes on the system run queue. If this is greater than the number of CPU's on the system, then excess load exists.

Note: For all target versions, the collection frequency for each metric is every 5 minutes.

The data sources for this metric category include the following:

Host	Data Source
Solaris	kernel statistics
HP	pstat_getdynamic(), pstat_getprocessor(), pstat_getproc(), pstat_getstatic(), getutent(), pstat_getvminfo() system calls
Linux	uptime, free, getconf, ps, iostat, sar, w OS commands; /proc/stat
HP Tru64	table() system call, uptime, vmstat, psrinfo, ps, who, swapon OS commands
IBM AIX	oracle_kstat(), getutent(), getproc(), sysconf() system calls
Windows	performance data counters (unless noted) (unless otherwise noted)

The following table lists the metrics and their descriptions.

Table 2–21 Load Metrics

Metric	Description
CPU in IO-Wait (%)	See Section 2.16.1, "CPU in IO-Wait (%)"
CPU in System Mode (%)	For UNIX-based platforms, this metric represents the amount of CPU being used in SYSTEM mode as a percentage of total CPU processing power. For Windows, this metric represents the percentage of time the process threads spent executing code in privileged mode.
CPU in User Mode (%)	For UNIX-based platforms, this metric represents the amount of CPU being used in USER mode as a percentage of total CPU processing power. For Windows, this metric represents the percentage of time the processor spends in the user mode. This metric displays the average busy time as a percentage of the sample time.
CPU Interrupt Time (%)	See Section 2.16.2, "CPU Interrupt Time (%)". **Note:** This metric is available only on Windows.
CPU Queue Length	See Section 2.16.3, "CPU Queue Length". **Note:** This metric is available only on Windows.
CPU Utilization (%)	See Section 2.16.4, "CPU Utilization (%)"
Free Memory (%)	Amount of free memory as a percentage of total memory. The data source for Windows host is Windows API.
Longest Service Time (ms)	Maximum of the average service time of all disks. Units are represented in milliseconds. **Note:** This metric is not available on Windows.
Memory Page Scan Rate (per second)	See Section 2.16.5, "Memory Page Scan Rate (per second)"
Memory Utilization (%)	See Section 2.16.6, "Memory Utilization (%)"
Page Transfers Rate	See Section 2.16.7, "Page Transfers Rate". **Note:** This metric is available only on Windows.
Run Queue Length (1 minute average)	See Section 2.16.8, "Run Queue Length (1 minute average)". **Note:** This metric is not available on Windows.
Run Queue Length (5 minute average)	See Section 2.16.10, "Run Queue Length (5 minute average)". **Note:** This metric is not available on Windows.
Run Queue Length (15 minute average)	See Section 2.16.9, "Run Queue Length (15 minute average)". **Note:** This metric is not available on Windows.
Swap Utilization (%)	See Section 2.16.11, "Swap Utilization (%)"

Table 2–21 (Cont.) Load Metrics

Metric	Description
Total Disk I/O Per Second	Rate of I/O (read and write) operations, calculated from all disks. **Note:** This metric is not available on Windows.
Total Processes	Total number of processes currently running on the system.
Total Swap, Kilobytes	Total amount of page file space available to be allocated by processes. Paging files are shared by all processes and the lack of space in paging files can prevent processes from allocating memory. **Note:** This metric is available only on Windows. The data sources for this metric are Performance Data counters and Windows API GlobalMemoryStatusEx.
Total Users	Represents the total number of users currently logged into the system. This metric checks the number of users running on the system. **Note:** This metric is not available on Windows.
Used Swap, Kilobytes	Size in kilobytes of the page file instance used. **Note:** This metric is available only on Windows. The data sources for this metric are Performance Data counters and Windows API GlobalMemoryStatusEx.

2.16.1 CPU in IO-Wait (%)

Represents the average number of jobs waiting for I/O in the last interval.

Metric Summary

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

Table 2–22 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	40	80	6	CPU I/O Wait is %value%%%, crossed warning (%warning_ threshold%) or critical (%critical_threshold%) threshold.

User Action

A high percentage of I/O wait can indicate a hardware problem, a slow NFS server, or poor load-balancing among local file systems and disks. Check the system messages log for any hardware errors. Use the iostat -xn command or the nfsstat -c (NFS client-side statistics) command or both to determine which disks or file systems are slow to respond. Check to see if the problem is with one or more swap partitions, as lack of swap or poor disk load balancing can cause these to become overloaded. Depending on the specific problem, fixes may include: NFS client or server tuning, hardware replacement, moving applications to other file systems, adding swap space, or restructuring a file system for better performance.

2.16.2 CPU Interrupt Time (%)

Represents the percentage of time the processor spends receiving and servicing hardware interrupts during sample intervals. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended during interrupts. Most system clocks interrupt the processor every 10 milliseconds, creating a background of interrupt activity. Suspends normal thread execution during interrupts.

This metric is available only on Windows.

Data Source

The data sources for this metric are Performance Data counters.

2.16.3 CPU Queue Length

Processor Queue Length is the number of ready threads in the processor queue. There is a single queue for processor time even on computers with multiple processors. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent on the workload.

This metric is available only on Windows.

Data Source

The data sources for this metric are Performance Data counters.

User Action

A consistently high value indicates a number of CPU bound tasks. This information should be corelated with other metrics such as Page Transfer Rate. Tuning the system, accompanied with additional memory, should help.

2.16.4 CPU Utilization (%)

For UNIX-based platforms, this metric represents the amount of CPU utilization as a percentage of total CPU processing power available.

For Windows, this metric represents the percentage of time the CPU spends to execute a non-Idle thread. CPU Utilization (%) is the primary indicator of processor activity.

Metric Summary

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

Table 2–23 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	80	95	6	CPU Utilization is %value%%%, crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

2.16.5 Memory Page Scan Rate (per second)

For UNIX-based systems, this metric represents the number of pages per second scanned by the page stealing daemon.

For Windows, this metric represents the rate at which pages are read from or written to disk to resolve hard page faults. The metric is a primary indicator of the kinds of faults that cause system-wide delays.

Metric Summary

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

Table 2–24 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	6	Page scan rate is %value% /sec, crossed warning (%warning_ threshold% /sec) or critical (%critical_threshold% /sec) threshold.

User Action

If this number is zero or close to zero, then you can be sure the system has sufficient memory. If scan rate is always high, then adding memory will definitely help.

2.16.6 Memory Utilization (%)

Represents the amount of free memory as a percentage of total memory.

Metric Summary

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

Table 2–25 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	99	Not Defined	6	Memory Utilization is %value%%%, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Data Source

For the Windows host, the data source is the Windows API.

2.16.7 Page Transfers Rate

Indicates the rate at which pages are read from or written to disk to resolve hard page faults. It is a primary indicator of the kinds of faults that cause systemwide delays. It is counted in numbers of pages. It includes pages retrieved to satisfy faults in the file system cache (usually requested by applications) non-cached mapped memory files.

This metric is available only on Windows.

Data Source

The data sources for this metric are Windows Performance counters.

User Action

High transfer rates indicate a memory contention. Adding memory would help.

2.16.8 Run Queue Length (1 minute average)

Represents the average number of processes in memory and subject to be run in the last interval. This metric checks the run queue.

This metric is not available on Windows.

User Action

Check the load on the system using the UNIX uptime or top commands. Also, check for processes using too much CPU time by using the top and ps -ef commands. Note that the issue may be a large number of instances of one or more processes, rather than a few processes each taking up a large amount of CPU time. Kill processes using excessive CPU time.

2.16.9 Run Queue Length (15 minute average)

Represents the average number of processes in memory and subject to be run in the last interval. This metric checks the run queue.

This metric is not available on Windows.

User Action

Check the load on the system using the UNIX uptime or top commands. Also, check for processes using too much CPU time by using the top and ps -ef commands. Note that the issue may be a large number of instances of one or more processes, rather than a few processes each taking up a large amount of CPU time. Kill processes using excessive CPU time.

2.16.10 Run Queue Length (5 minute average)

Represents the average number of processes in memory and subject to be run in the last interval. This metric checks the run queue.

This metric is not available on Windows.

Metric Summary

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

Table 2–26 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	10	20	6	CPU Load is %value%, crossed warning (%warning_ threshold%) or critical (%critical_threshold%) threshold.

User Action

Check the load on the system using the UNIX uptime or top commands. Also, check for processes using too much CPU time by using the top and ps -ef commands. Note that the issue may be a large number of instances of one or more processes, rather than a few processes each taking up a large amount of CPU time. Kill processes using excessive CPU time.

2.16.11 Swap Utilization (%)

For UNIX-based platforms, this metric represents the percentage of swapped memory in use for the last interval.

For Windows, this metric represents the percentage of page file instance used.

Metric Summary

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

Table 2–27 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	80	95	6	Swap Utilization is %value%%%, crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

Data Source

The data sources for the Windows host are Windows API and performance data counters.

User Action

For UNIX-based platforms, check the swap usage using the UNIX top command or the Solaris swap -l command. Additional swap can be added to an existing file system by creating a swap file and then adding the file to the system swap pool. (See documentation for your UNIX OS). If swap is mounted on /tmp, space can be freed by removing any junk files in /tmp. If it is not possible to add file system swap or free up enough space, additional swap will have to be added by adding a raw disk partition to the swap pool. See UNIX documentation for procedures.

For Windows, check the page file usage and add an additional page file if current limits are insufficient.

2.17 Log File Monitoring

The Log File Monitoring metric allows the operator to monitor one or more log files for the occurrence of one or more perl patterns in the content. In addition, the operator can specify a perl pattern to be ignored for the log file. Periodic scanning will be performed against new content added since the last scan, lines matching the *ignore pattern* will be ignored first, then lines matching specified *match patterns* will result in one record being uploaded to the repository for each pattern. The user can set a threshold against the number of lines matching the given pattern. File rotation will be handled within the given file.

2.17.1 Log File Pattern Matched Content

Returns the actual content if the given file has been specifically registered for content uploading, else it will return the count of lines that matched the pattern specified.

The operator can list the names of files or directories to be never monitored in <EMDROOT>/sysman/config/lfm_efiles file. The operator can list the names of the files or directories whose contents can be uploaded into Oracle Management Repository in <EMDROOT>/sysman/config/lfm_ifiles file.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 15 Minutes

Data Source

Oracle provided perl program that scans files for the occurrence of user specified perl patterns.

2.17.2 Log File Pattern Matched Line Count

Returns the number of lines matching the pattern specified in the given file. Setting warning or critical thresholds against this column for a specific {log file name, match pattern in perl, ignore pattern in perl} triggers the monitoring of specified criteria against the given log file.

Metric Summary

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

Table 2–28 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every Sample	>	0	Not Defined	1*	%log_file_message% Crossed warning (%warning_ threshold%) or critical (%critical_threshold%) threshold.

* Once an alert is triggered for this metric, it must be manually cleared.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Log File Name", "Match Pattern in Perl", "Ignore Pattern in Perl", and "Time Stamp" objects.

If warning or critical threshold values are currently set for any unique combination of "Log File Name", "Match Pattern in Perl", "Ignore Pattern in Perl", and "Time Stamp" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Log File Name", "Match Pattern in Perl", "Ignore Pattern in Perl", and "Time Stamp" objects, use the Edit Thresholds page.

Data Source

Oracle supplied perl program monitors the log files for user specified criteria.

2.18 Memory Devices

The Memory Devices metric monitors the status of memory devices configured in the system.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the metrics, descriptions, and data sources.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

Table 2–29 Memory Devices Metrics

Metric	Description	Data Source (SNMP MIB Object)
Bank Location	Bank location name of the memory device, when applicable	memoryDeviceBankLocationName (1.3.6.1.4.1.674.10892.1.1100.50.1.10)
Location	Location name of the memory device, for example, "DIMM A".	memoryDeviceLocationName (1.3.6.1.4.1.674.10892.1.1100.50.1.8)
Memory	See Section 2.18.1, "Memory Status"	Section 2.18.1, "Memory Status"
Size (MB)	Size, in kilobytes, of the memory device	memoryDeviceSize (1.3.6.1.4.1.674.10892.1.1100.50.1.14)

Table 2–29 (Cont.) Memory Devices Metrics

Metric	Description	Data Source (SNMP MIB Object)
Type	Type of the memory device	memoryDeviceType (1.3.6.1.4.1.674.10892.1.1100.50.1.7)

2.18.1 Memory Status

Represents the status of the memory device.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–30 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of Memory at bank location %MemoryBankLocation% and location %MemoryLocation% is %value%, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis" and "Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis" and "Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis" and "Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: memoryDeviceStatus (1.3.6.1.4.1.674.10892.1.1100.50.1.5)

2.19 Message and Semaphore Activity

The Message and Semaphore Activity metric provides information about the message and semaphore activity of the host system being monitored.

The data sources for this metric include the following:

Host	Data Source
Solaris	sar command
HP	sar command
Linux	not available
HP Tru64	ipcs command
IBM AIX	sar command
Windows	not available

The following table lists the metrics and their descriptions.

Table 2–31 Message and Semaphore Activity

Metric	Description
msgrcv() System Calls (per second)	Number of msgrcv system calls made per second. The msgrcv system call reads a message from one queue to another user-defined queue.
semop() System Calls (per second)	Number of semop system calls made per second. The semop system call is used to perform semaphore operations on a set of semaphores.

2.20 Network Interfaces

The Network Interfaces metric includes input errors and interface collisions on the network interface. The following network interfaces are supported: le, hme, qfe, ge, and fddi.

Note: For all target versions, the collection frequency for each metric is every 5 minutes.

Data Source

The data sources for the metrics in this category include the following:

Host	Data Source
Solaris	kernel memory structures (kstat)
HP	netstat, lanscan, and lanadmin commands
Linux	netstat command and /proc/net/dev
HP Tru64	netstat command
IBM AIX	oracle_kstat() system call

Host	Data Source
Windows	not available

User Action

Use the OS netstat -i command to check the performance of the interface. Also, check the system messages file for messages relating to duplex setting by using the OS grep -i command and searching for the word 'duplex'.

Metrics and Descriptions

The following table lists the metrics and their descriptions.

Table 2–32 Network Interfaces Metrics

Metric	Description
Network Interface Input Errors (%)	Number of input errors, per second, encountered on the device for unsuccessful reception due to hardware/network errors. This metric checks the rate of input errors on the network interface specified by the network device names parameter, such as le0 or * (for all network interfaces).
Network Interface Collisions (%)	Number of collisions per second. This metric checks the rate of collisions on the network interface specified by the network device names parameter, such as le0 or * (for all network interfaces).
Network Interface Combined Utilization (%)	See Section 2.20.1, "Network Interface Combined Utilization (%)"
Network Interface Output Errors (%)	Number of output errors per second. This metric checks the rate of output errors on the network interface specified by the network device names parameter, such as le0 or * (for all network interfaces).
Network Interface Read (MB/s)	Amount of megabytes per second read from the specific interface
Network Interface Read Utilization (%)	Amount of network bandwidth being used for reading from the network as a percentage of total read capacity
Network Interface Total Error Rate (%)	See Section 2.20.2, "Network Interface Total Error Rate (%)"
Network Interface Total I/O Rate (MB/sec)	See Section 2.20.3, "Network Interface Total I/O Rate (MB/sec)"
Network Interface Write (MB/s)	Amount of megabytes per second written to the specific interface
Network Interface Write Utilization (%)	Amount of network bandwidth being used for writing to the network as a percentage of total read capacity.

2.20.1 Network Interface Combined Utilization (%)

Represents the percentage of network bandwidth being used by reading and writing from and to the network for full-duplex network connections.

Metric Summary

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

Table 2–33 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	6	Network utilization for %keyvalue% is %value%%%, crossed warning (%warning_ threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Network Interface Name" object.

If warning or critical threshold values are currently set for any "Network Interface Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Network Interface Name" object, use the Edit Thresholds page.

2.20.2 Network Interface Total Error Rate (%)

Represents the number of total errors per second, encountered on the network interface. It is the rate of read and write errors encountered on the network interface.

Metric Summary

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

Table 2–34 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	6	Network Error Rate for %keyvalue% is %value%%%, crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Network Interface Name" object.

If warning or critical threshold values are currently set for any "Network Interface Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Network Interface Name" object, use the Edit Thresholds page.

Data Source

It is computed as the sum of Network Interface Input Errors (%) and Network Interface Output Errors (%).

2.20.3 Network Interface Total I/O Rate (MB/sec)

Represents the total I/O rate on the network interface. It is measured as the sum of Network Interface Read (MB/s) and Network Interface Write (MB/s).

Metric Summary

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

Table 2–35 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	6	Network I/O Rate for %keyvalue% is %value%MB/Sec, crossed warning (%warning_ threshold%MB/Sec) or critical (%critical_threshold%MB/Sec) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each "Network Interface Name" object.

If warning or critical threshold values are currently set for any "Network Interface Name" object, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each "Network Interface Name" object, use the Edit Thresholds page.

Data Source

It is computed as the sum of Network Interface Read (MB/s) and Network Interface Write (MB/s).

2.21 Paging Activity

The Paging Activity metric provides the amount of paging activity on the system.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

Data Source

The data sources for this metric category include the following:

Host	Data Source
Solaris	kernel statistics (class misc cpu_stat)
HP	pstat_getvminfo() system call
Linux	sar command
HP Tru64	table(() system call and vmstat command
IBM AIX	oracle_kstat() system call
Windows	performance data counters

Metrics and Descriptions

The following table lists the metrics and their descriptions:

Table 2–36 Paging Activity Metrics

Metric	Description
Address Translation Page Faults (per second)	Minor page faults by way of hat_fault() per second. This metric checks the number of faults for the CPU(s) specified by the Host CPU(s) parameter, such as cpu_stat0 or * (for all CPUs on the system). **Note:** This metric is not available on Linux and Windows.
Cache Faults	Rate at which faults occur when a page sought in the file system cache is not found and must be retrieved from elsewhere in memory (a soft fault) or from disk (a hard fault). The file system cache is an area of physical memory that stores recently used pages of data for applications. Cache activity is a reliable indicator of most application I/O operations. This metric shows the number of faults, without regard for the number of pages faulted in each operation. **Note:** This metric is available only on Windows.
Copy-on-write Faults (per second)	Rate at which page faults are caused by attempts to write that have been satisfied by coping of the page from elsewhere in physical memory. This is an economical way of sharing data since pages are only copied when they are written to; otherwise, the page is shared. This metric shows the number of copies, without regard for the number of pages copied in each operation. **Note:** This metric is available only on Windows.
Demand Zero Faults (per second)	Rate at which a zeroed page is required to satisfy the fault. Zeroed pages, pages emptied of previously stored data and filled with zeros, are a security feature of Windows that prevent processes from seeing data stored by earlier processes that used the memory space. Windows maintains a list of zeroed pages to accelerate this process. This metric shows the number of faults, without regard to the number of pages retrieved to satisfy the fault. **Note:** This metric is available only on Windows.
igets with Page Flushes (%)	Represents the percentage of UFS inodes taken off the freelist by iget which had reusable pages associated with them. These pages are flushed and cannot be reclaimed by processes. **Note:** This metric is available on Solaris, HP, and IBM AIX.
Page Faults (per second)	Average number of pages faulted per second. It is measured in number of pages faulted per second because only one page is faulted in each fault operation, hence this is also equal to the number of page fault operations. This metric includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory.) Most processors can handle large numbers of soft faults without significant consequence. However, hard faults, which require disk access, can cause significant delays. **Note:** This metric is available only on Windows.

Table 2–36 (Cont.) Paging Activity Metrics

Metric	Description
Page Faults from Software Lock Requests	Represents the number of protection faults per second. These faults occur when a program attempts to access memory it should not access, receives a segmentation violation signal, and dumps a core file. This metric checks the number of faults for the CPU(s) specified by the Host CPU(s) parameter, such as cpu_stat0 or * (for all CPUs on the system). **Note:** This metric is not available on Linux or Windows.
Page-in Requests (per second)	For UNIX-based systems, represents the number of page read ins per second (read from disk to resolve fault memory references) by the virtual memory manager. Along with Page Outs, this statistic represents the amount of real I/O initiated by the virtual memory manager. This metric checks the number of page read ins for the CPU(s) specified by the Host CPU(s) parameter, such as cpu_stat0 or * (for all CPUs on the system). For Windows, this metric is the rate at which the disk was read to resolve hard page faults. It shows the number of reads operations, without regard to the number of pages retrieved in each operation. Hard page faults occur when a process references a page in virtual memory that is not in working set or elsewhere in physical memory, and must be retrieved from disk. This metric is a primary indicator of the kinds of faults that cause systemwide delays. It includes read operations to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. **Note:** This metric is not available on Linux.
Page-out Requests (per second)	For UNIX-based systems, represents the number of page write outs to disk per second. This metric checks the number of page write outs for the CPU(s) specified by the Host CPU(s) parameter, such as cpu_stat0 or * (for all CPUs on the system). For Windows, this metric is the rate at which pages are written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. This metric shows write operations, without regard to the number of pages written in each operation. **Note:** This metric is not available on Linux.
Pages Paged-in (per second)	For UNIX-based systems, represents the number of pages paged in (read from dirk to resolve fault memory references) per second. This metric checks the number of pages paged in for the CPU(s) specified by the Host CPU(s) parameter, such as cpu_stat0 or * (for all CPUs on the system). For Windows, this metric is the rate at which pages are read from disk to resolve hard page faults. Hard page faults occur when a process refers to a page in virtual memory that is not in its working set or elsewhere in physical memory, and must be retrieved from disk. When a page is faulted, the system tries to read multiple contiguous pages into memory to maximize the benefit of the read operation.
Pages Paged-out (per second)	For UNIX-based systems, represents the number of pages written out (per second) by the virtual memory manager. Along with Page Outs, this statistic represents the amount of real I/O initiated by the virtual memory manager. This metric checks the number of pages paged out for the CPU(s) specified by the Host CPU(s) parameter, such as cpu_stat0 or * (for all CPUs on the system). For Windows, this metric is the rate at which pages are written to disk to free up space in physical memory. Pages are written back to disk only if they are changed in physical memory, so they are likely to hold data, not code. A high rate of pages output might indicate a memory shortage. Windows writes more pages back to disk to free up space when physical memory is in short supply.

Table 2–36 (Cont.) Paging Activity Metrics

Metric	Description
Pages Put on Freelist by Page Stealing Daemon (per second)	Number of pages that are determined unused, by the pageout daemon (also called the page stealing daemon), and put on the list of free pages. **Note:** This metric is not available on Linux and Windows.
Pages Scanned by Page Stealing Daemon (per second)	Represents the scan rate is the number of pages per second scanned by the page stealing daemon. If this number is zero or closer to zero, then you can be sure the system has sufficient memory. If the number is always high, then adding memory will definitely help. **Note:** This metric is not available on Linux and Windows.
Transition Faults (per second)	Rate at which page faults are resolved by recovering pages that were being used by another process sharing the page, or were on the modified page list or the standby list, or were being written to disk at the time of the page fault. The pages were recovered without additional disk activity. Transition faults are counted in numbers of faults; because only one page is faulted in each operation, it is also equal to the number of pages faulted. **Note:** This metric is available only on Windows.

2.22 PCI Devices

The Peripheral Component Interconnect (PCI) Devices metric monitors the status of PCI devices.

This metric is available only on Dell Poweredge Linux Systems.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The following table lists the metrics, their descriptions, and user actions.

Table 2–37 PCI Devices Metrics

Metric	Description	Data Source (SNMP MIB Object)
Description	Descriptive name of the Dell Peripheral Component Interconnect (PCI) Device	pCIDeviceDescriptionName (1.3.6.1.4.1.674.10892.1.1100.80.1.9)
Manufacturer	Name of the Dell Peripheral Component Interconnect (PCI) Device manufacturer	pCIDeviceManufacturerName (1.3.6.1.4.1.674.10892.1.1100.80.1.8)
PCI Device Status	See Section 2.22.1, "PCI Device Status"	See Section 2.22.1, "PCI Device Status"

2.22.1 PCI Device Status

Represents the status of the Dell Peripheral Component Interconnect (PCI) Device.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)

Metric Value	Meaning (per SNMP MIB)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–38 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of PCIDevice %PCIDeviceIndex% in chassis %ChassisIndex% is %value%, crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis Index", "PCI Device Index", and "System Slot Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis Index", "PCI Device Index", and "System Slot Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis Index", "PCI Device Index", and "System Slot Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: pCIDeviceStatus (1.3.6.1.4.1.674.10892.1.1100.80.1.5)

2.23 Power Supplies

The Power Supplies metric monitors the status of various power supplies present in the host system.

This metric is available only on Dell Poweredge Linux Systems.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The following table lists the metrics, their descriptions, and user actions.

Table 2–39 Power Supplies Metrics

Metric	Description	Data Source (SNMP MIB Object)
Location	Location name of the power supply	powerSupplyLocationName (1.3.6.1.4.1.674.10892.1.600.12.1.8
Output (Tenths of Watts)	maximum sustained output wattage of the power supply, in tenths of watts	powerSupplyOutputWatts (1.3.6.1.4.1.674.10892.1.600.12.1.6)
Power Supply Status	See Section 2.23.1, "Power Supply Status"	See Section 2.23.1, "Power Supply Status"

2.23.1 Power Supply Status

Represents the status of the power supply.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–40 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of Power Supply %PSIndex% in chassis %ChassisIndex% is %value%, crossed warning (%warning_ threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis Index" and "Power Supply Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis Index" and "Power Supply Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis Index" and "Power Supply Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: powerSupplyStatus (1.3.6.1.4.1.674.10892.1.600.12.1.5)

2.24 Process, Inode, File Tables Statistics

The Process, Inode, File Tables Stats metric provides information about the process, inode, and file tables status.

Data Source

The data sources for this metric category include the following:

Host	Data Source
Solaris	sar command
HP	sar command
Linux	sar command, for example, sar -v
HP Tru64	table() system call
IBM AIX	sar command
Windows	not available

The OS sar command is used to sample cumulative activity counters maintained by the OS. The data is obtained by sampling system counters once in a five-second interval.

Metrics and Descriptions

The following table lists the metrics and their descriptions.

Table 2–41 Process, Inode, File Tables Statistics Metrics

Metric	Description
File Table Overflow Occurrences	Number of times the system file table overflowed, that is, the number of times that the OS could not find any available entries in the table in the sampling period chosen to collect the data. **Note:** This metric is not available on Linux or Windows.
Inode Table Overflow Occurrences	Number of times the inode table overflowed, that is, the number of times the OS could not find any available inode table entries. **Note:** This metric is not available on Linux or Windows.
Maximum Size of Inode Table	Maximum size of the inode table. **Note:** This metric is not available on Linux or Windows.
Maximum Size of Process Table	Maximum size of the process table. **Note:** This metric is not available on Linux or Windows.
Maximum Size of System File Table	Maximum size of the system file table. **Note:** This metric is not available on Linux or Windows.
Number of Allocated Disk Quota Entries	Number of allocated disk quota entries. **Note:** This metric is available only on Linux.
Number of Queued RT Signals	Number of queued RT signals. **Note:** This metric is available only on Linux.

Table 2–41 (Cont.) Process, Inode, File Tables Statistics Metrics

Metric	Description
Number of Super Block Handlers Allocated	Number of allocated super block handlers. **Note:** This metric is available only on Linux.
Number of Used File Handles	Current size of the system file table.
Percentage of Allocated Disk Quota Entries	Percentage Of Allocated Disk Quota Entries against the maximum number of cached disk quota entries that can be allocated. **Note:** This metric is available only on Linux.
Percentage of Allocated Super Block Handlers	Percentage Of Allocated Super Block Handlers against the maximum number of super block handlers that Linux can allocate. **Note:** This metric is available only on Linux.
Percentage of Queued RT Signals	Percentage of queued RT signals. **Note**: This metric is available only on Linux.
Percentage of Used File Handles	Percentage of used file handles against the maximum number of file handles that the Linux kernel can allocate. **Note:** This metric is available only on Linux.
Process Table Overflow Occurrences	Number of times the process table overflowed, that is, the number of times the OS could not find any process table entries in a five-second interval. **Note:** This metric is not available on Linux or Windows.
Size of Inode Table	Current size of the inode table.
Size of Process Table	Current size of the process table. **Note:** This metric is not available on Linux or Windows

2.25 Processors

The Processors metric monitors the state of each CPU in the host.

This metric is available only on Dell Poweredge Linux Systems.

> **Note:** For all target versions, the collection frequency for each metric is every 15 minutes.

The following table lists the metrics, descriptions, and data sources.

Table 2–42 Processors Metrics

Metric	Description	Data Source (SNMP MIB Object)
Family	Family of the Dell process device	processorDeviceFamily (1.3.6.1.4.1.674.10892.1.1100.30.1.10)
Manufacturer	Name of the manufacturer of the Dell processor	processorDeviceManufacturerName (1.3.6.1.4.1.674.10892.1.1100.30.1.8)
Processor Status	See Section 2.25.1, "Processor Status"	See Section 2.25.1, "Processor Status"
Speed (MHz)	current speed of the Dell processor device in Mega Hertz (MHz). A value of zero indicates the speed is unknown.	processorDeviceCurrentSpeed (1.3.6.1.4.1.674.10892.1.1100.30.1.12)
Version	Version of the Dell processor	processorDeviceVersionName (1.3.6.1.4.1.674.10892.1.1100.30.1.16)

2.25.1 Processor Status

Represents the status of the Dell processor device.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–43 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of Processor %ProcessorIndex% in chassis %ChassisIndex% is %value%, crossed warning (%warning_ threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis Index" and "Processor Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis Index" and "Processor Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis Index" and "Processor Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: processorDeviceStatus (1.3.6.1.4.1.674.10892.1.1100.30.1.5)

2.26 Program Resource Utilization

The Program Resource Utilization metric provides flexible resource monitoring functionality. The operator must specify the criteria for the programs to be monitored by specifying key value specific thresholds. Values for the key value columns

{program name, owner} define the unique criteria to be monitored for resource utilization in the system.

By default, no programs will be tracked by this metric. Key Values entered as part of a key value specific threshold setting define the criteria for monitoring and tracking.

Note: For all target versions, the collection frequency for each metric is every 5 minutes.

The data sources for this metric category include the following:

Host	Data Source
Solaris	ps command
HP	ps command
Linux	ps command
HP Tru64	ps command
IBM AIX	ps command
Windows	performance data counters

The following table lists the metrics and their descriptions.

Table 2–44 Program Resource Utilization Metrics

Metric	Description
List of PIDs	This metric is only available on Solaris.
Program's Max CPU Time Accumulated (Minutes)	See Section 2.26.1, "Program's Max CPU Time Accumulated (Minutes)"
Program's Max CPU Time Accumulated PID	Identifier of the process that has accumulated the most CPU time matching the {program name, owner} key value criteria
Program's Max CPU Utilization (%)	See Section 2.26.2, "Program's Max CPU Utilization (%)"
Program's Max CPU Utilization PID	Identifier of the process with the maximum percentage of CPU utilized matching the {program name, owner} key value criteria since last scan
Program's Max Process Count	See Section 2.26.3, "Program's Max Process Count"
Program's Max Resident Memory (MB)	See Section 2.26.4, "Program's Max Resident Memory (MB)"
Program's Max Resident Memory PID	Identifier of the process with the maximum resident memory occupied by a single process matching the {program name, owner} key value criteria
Program's Min Process Count	See Section 2.26.5, "Program's Min Process Count"
Program's Total CPU Time Accumulated (Minutes)	See Section 2.26.6, "Program's Total CPU Time Accumulated (Minutes)"
Program's Total CPU Utilization (%)	See Section 2.26.7, "Program's Total CPU Utilization (%)"

2.26.1 Program's Max CPU Time Accumulated (Minutes)

Represents the maximum CPU time accumulated by the most active process matching the {program name, owner} key value criteria.

Metric Summary

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

Table 2–45 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	3	%prog_max_cpu_time_pid% process running program %prog_name% has accumulated %prog_max_cpu_time% minutes of cpu time. This duration crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.26.2 Program's Max CPU Utilization (%)

Represents the maximum percentage of CPU utilized by a single process matching the {program name, owner} key value criteria since last scan.

Metric Summary

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

Table 2–46 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	3	Process %prog_max_cpu_util_pid% running program %prog_name% is utilizing %prog_max_cpu_util%% cpu. This percentage crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.26.3 Program's Max Process Count

Fetches the current number of processes matching the {program name, owner} key value criteria. It can be used for setting warning or critical thresholds to monitor for maximum number of processes that a given {program name, owner} key value criteria crosses.

Metric Summary

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

Table 2–47 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	3	%prog_max_process_count% processes are running program %prog_name% owned by [%owner%], crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.26.4 Program's Max Resident Memory (MB)

Represents the maximum resident memory occupied by a single process matching the {program name, owner} key value criteria. It can be used for setting warning or critical thresholds to monitor for maximum value a given {program name, owner} key value criteria crosses.

Metric Summary

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

Table 2–48 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	3	%prog_max_rss_pid% process running program %prog_name% is utilizing %prog_max_rss% (MB) of resident memory. This percentage crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.26.5 Program's Min Process Count

Fetches the current number of processes matching the {program name, owner} key value criteria. It can be used for setting warning or critical thresholds to monitor for minimum number of processes that a given {program name, owner} key value criteria should never go under.

Metric Summary

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

Table 2–49 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	<	Not Defined	Not Defined	3	%prog_max_process_count% processes are running program %prog_name% owned by [%owner%], fallen below warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.26.6 Program's Total CPU Time Accumulated (Minutes)

Represents the total CPU time accumulated by all active process matching the {program name, owner} key value criteria.

Metric Summary

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

Table 2–50 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	3	%prog_max_count% processes running program %prog_name% owned by [%owner%] have accumulated %prog_total_cpu_time% minutes of cpu time. This duration crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.26.7 Program's Total CPU Utilization (%)

Represents the percentage of CPU time utilized by all active process matching the {program name, owner} key value criteria since last collection.

Metric Summary

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

Table 2–51 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	After Every Sample	>	Not Defined	Not Defined	3	%prog_max_count% processes running program %prog_name% owned by [%owner%] are utilizing %prog_total_cpu_ util%% cpu. This percentage crossed warning (%warning_ threshold%) or critical (%critical_ threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Program Name" and "Owner" objects.

If warning or critical threshold values are currently set for any unique combination of "Program Name" and "Owner" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Program Name" and "Owner" objects, use the Edit Thresholds page.

2.27 Remote Access Card

The Remote Access Card metric monitors the status of the Remote Access Card.

This metric is available only on Dell Poweredge Linux Systems.

> **Note:** For all target versions, the collection frequency for each metric is every 15 minutes.

The following table lists the metrics, their descriptions, and data sources.

Table 2–52 Remote Access Card Metrics

Metric	Description	Data Source (SNMP MIB Object)
DHCP Settings	Determines whether the dynamic host configuration protocol (DHCP) was used to obtain the network interface card (NIC) information.	remoteAccessNICCurrentInfoFromDHCP (1.3.6.1.4.1.674.10892.1.1700.10.1.33)
Gateway Address	Represents the IP address for the gateway currently being used by the onboard network interface card (NIC) provided by the remote access (RAC) hardware.	remoteAccessNICCurrentGatewayAddress (1.3.6.1.4.1.674.10892.1.1700.10.1.32)
IP Address	Provides the internet protocol (IP) address currently being used by the onboard network interface card (NIC) provided by the remote access (RAC) hardware	remoteAccessNICCurrentIPAddress (1.3.6.1.4.1.674.10892.1.1700.10.1.30)
LAN Settings	Represents the local area network (LAN) settings of the remote access hardware.	remoteAccessLANSettings (1.3.6.1.4.1.674.10892.1.1700.10.1.15
Network Mask Address	Represents the subnet mask currently being used by the onboard network interface card (NIC) provided by the remote access (RAC) hardware.	remoteAccessNICCurrentNetmaskAddress (1.3.6.1.4.1.674.10892.1.1700.10.1.31)
Product Name	Represents the name of the product providing the remote access (RAC) functionality	remoteAccessProductInfoName (1.3.6.1.4.1.674.10892.1.1700.10.1.7)
Remote Access Card State	Represents the state of the remote access (RAC) hardware.	remoteAccessStateSettings (1.3.6.1.4.1.674.10892.1.1700.10.1.5)
Remote Access Card Status	See Section 2.27.1, "Remote Access Card Status"	See Section 2.27.1, "Remote Access Card Status"
Version	Represents the version of the product providing the remote access (RAC) functionality.	remoteAccessVersionInfoName (1.3.6.1.4.1.674.10892.1.1700.10.1.9)

2.27.1 Remote Access Card Status

Represents the status of the remote access (RAC) hardware.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical

Metric Value	Meaning (per SNMP MIB)
6	Non-Recoverable

Metric Summary

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

Table 2–53 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of Remote Access Card is %value%, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Data Source

SNMP MIB object: remoteAccessStatus (1.3.6.1.4.1.674.10892.1.1700.10.1.6)

2.28 Response

This metric provides the status of the host, that is, whether it is up or down.

2.28.1 Status

The metric indicates whether the host is reachable or not. A host could be unreachable due to various reasons. The network is down or the Management Agent on the host is down (which could be because the host itself is shutdown).

2.29 Storage Summary Metrics

The Storage Summary metrics collectively represent the summary of storage data on a host target. These metrics are derived from the various metrics collected and uploaded into the Oracle Management Repository by the Management Agent. They are computed every time the Management Agent populates the Management Repository with storage data. This collection is also triggered automatically whenever the user manually refreshes the host storage data from the Storage Details page.

These metrics are available on the Linux and Solaris hosts.

Note: For target versions 3.0 and higher, the collection frequency for each metric is every 24 hours or when the user manually refreshes storage data from the Storage Details page.

For more details on how these metrics are computed see the "About Storage Computation Formulas" topic in the Enterprise Manager online help. The online help also provides information about ASM, databases, disks, file systems, volumes, and storage details.

The following table lists the metrics and their descriptions.

Table 2–54 Storage Summary Metrics

Metric	Description
ASM Storage Allocated (GB)	Total storage allocated to Oracle databases from Automatic Storage Management (ASM) instances on the host
ASM Storage Metric Collection Errors	Number of metric collection errors attributed to the storage related metrics of the Automatic Storage Management (ASM) targets on the host
ASM Storage Overhead (GB)	Storage overhead of Automatic Storage Management (ASM) targets on the host
ASM Storage Unallocated (GB)	Storage available in Automatic Storage Management (ASM) targets on the host for allocating to databases
Databases Storage Free (GB)	Total free storage available in the databases on the host
Databases Storage Metric Collection Errors	Metric collection errors of storage related metrics of databases on the host
Databases Storage Used (GB)	Total free storage available in the databases on the host
Disk Storage Allocated (GB)	Storage allocated from the total disk storage available on the host
Disk Storage Unallocated (GB)	Storage that is available for allocation in disks on the host.
Host Storage Metric Collection Errors	Total number of storage related metric collection errors of the host target
Hosts Summarized	The possible values for this metric are: ■ 1 (one) if this host storage was computed successfully (sometimes with partial errors) ■ 0 (zero) if the storage computation did not proceed at all due to some reasons (for example, failure to collect critical storage metric data).
Local File Systems Storage Free (GB)	Total free storage in all distinct local file systems on the host
Local File Systems Storage Used (GB)	Total used space in all distinct local file systems on the host
Number of ASM Instances Summarized	Total number of Automatic Storage Management (ASM) instances, the storage data of which was used in computing storage summary of this host
Number of Databases Summarized	Total number of databases, the storage data of which was used in computing storage summary of this host
Other Mapping Errors	Storage metric mapping issues on the host excluding the unmonitored server mapping errors
Total Number of ASM Instances	Total number of Automatic Storage Management (ASM) instances on the host
Total Number of Databases	Total number of databases on the host
Total Storage Allocated (GB)	Total storage allocated from the host-visible storage available on the host
Total Storage Free (GB)	Free storage available from the total allocated storage on the host

Table 2–54 (Cont.) Storage Summary Metrics

Metric	Description
Total Storage Overhead (GB)	Overhead associated with storage on the host
Total Storage Unallocated (GB)	Total unallocated storage on the host
Total Storage Used (GB)	Total storage used in the file systems and databases on the host
Unmonitored NFS Server Mapping Errors	Total number of storage mapping issues that result from unmonitored Network File Systems (NFS) servers
Volumes Storage Allocated (GB)	Total storage allocated from the volumes available on the host
Volumes Storage Overhead (GB)	Storage overhead in the volumes on the host
Volumes Storage Unallocated (GB)	Storage available for allocation in the volumes on the host
Writeable NFS Storage Free (GB)	Total free space available in all distinct writeable NFS mounts on the host
Writeable NFS Storage Used (GB)	Storage used in all writeable NFS mounts on the host

2.30 Swap Area Status

The Swap Area Status metric provides the status of the swap memory on the system.

The data sources for this metric category include the following:

Host	Data Source
Solaris	swap
HP	swapinfo
Linux	/proc/swaps
HP Tru64	swapon
IBM AIX	lsps
Windows	not available

2.30.1 Swap Free

Represents the number of 1K blocks in swap area that is not allocated.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 24 Hours

User Action

Check the swap usage using the UNIX top command or the Solaris swap -l command. Additional swap can be added to an existing file system by creating a swap file and then adding the file to the system swap pool. (See documentation for your UNIX OS).

If swap is mounted on /tmp, space can be freed by removing any junk files in /tmp. If it is not possible to add file system swap or free up enough space, additional swap will have to be added by adding a raw disk partition to the swap pool. See UNIX documentation for procedures.

2.30.2 Swap Size

Represents the size of the swap file.

Metric Summary

The following table shows how often the metric's value is collected.

Target Version	Collection Frequency
All Versions	Every 24 Hours

2.31 Switch/Swap Activity

The Switch/Swap Activity metric displays the metric reports on the system switching and swapping activity.

Data Source

The data sources for this metric category, unless otherwise stated, include the following:

Host	Data Source
Solaris	sar command
HP	sar command
Linux	sar command
HP Tru64	not available
IBM AIX	sar command
Windows	not available

The OS sar command is used to sample cumulative activity counters maintained by the OS. Also, the data is obtained by sampling system counters once in a five-second interval. The results are essentially the number of processes swapped in over this five-second period divided by five.

Metrics and Descriptions

The following table lists the metrics and their descriptions.

Table 2–55 Switch/Swap Activity Metrics

Metric	Description
Process Context Switches (per second)	Number of process context switches per second. **Note:** This metric is available on Solaris, HP, and IBM AIX.
Swapins Transfers (per second)	Number of 512-byte units transferred for swapins per second. **Note:** This metric is not available on HP Tru64.
Swapout Transfers (per second)	Number of 512-byte units transferred for swapouts per second. **Note:** This metric is not available on HP Tru64.

Table 2–55 (Cont.) Switch/Swap Activity Metrics

Metric	Description
System Swapins (per second)	Number of process swapins per second. **Note:** This metric is not available on HP Tru64.
System Swapouts (per second)	Number of process swapouts per second. **Note:** This metric is not available on HP Tru64

2.32 System BIOS

The System BIOS (Basic Input/Output System) metric monitors the BIOS status for Dell Poweredge Linux systems.

This metric is available only on Dell Poweredge Linux Systems.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The following table lists the metrics, their descriptions, and data sources.

Table 2–56 System BIOS Metrics

Metric	Description	Data Source (SNMP MIB Object)
Manufacturer	Manufacturer's name of the System BIOS (Basic Input/Output System	systemBIOSManufacturerName (1.3.6.1.4.1.674.10892.1.300.50.1.11
Size	Image size of the System BIOS (Basic Input/Output System) in kilobytes. A value of zero indicates that the size is unknown.	systemBIOSSize (1.3.6.1.4.1.674.10892.1.300.50.1.6)
System BIOS Status	See Section 2.32.1, "System BIOS Status"	See Section 2.32.1, "System BIOS Status"
Version	Version name of the System BIOS (Basic Input/Output System)	systemBIOSVersionName (1.3.6.1.4.1.674.10892.1.300.50.1.8)

2.32.1 System BIOS Status

Represents the status of the System BIOS (Basic Input/Output System) in this chassis.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–57 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Status of BIOS %BiosIndex% in chassis %ChassisIndex% is %value%, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis Index" and "System BIOS Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis Index" and "System BIOS Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis Index" and "System BIOS Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: systemBIOSStatus (1.3.6.1.4.1.674.10892.1.300.50.1.5)

2.33 System Calls

The System Calls metric provides statistics about the system calls made over a five-second interval.

Data Source

The data sources for this metric category, unless otherwise stated, include the following:

Host	Data Source
Solaris	sar command
HP	sar command
Linux	not available
HP Tru64	table(() system call
IBM AIX	sar command
Windows	not available

The OS sar command is used to sample cumulative activity counters maintained by the OS. The data is obtained by sampling system counters once in a five-second

interval. The results are essentially the number of system calls made over this period divided by the period.

Metrics and Descriptions

The following table lists the metrics and their descriptions.

Table 2–58 System Calls Metrics

Metric	Description
Characters Transferred by Read System Calls (per second)	Number of characters transferred by read system calls (block devices only) per second
Characters Transferred by Write System Calls (per second)	Number of characters transferred by write system calls (block devices only) per second
exec() System Calls (per second)	Number of exec() system calls made per second
fork() System Calls (per second)	Number of fork() system calls made per second
read() System Calls (per second)	Number of read() system calls made per second
System Calls (per second)	Number of system calls made per second. This includes system calls of all types.
write() System Calls (per second)	Number of write() system calls made per second

2.34 Temperature

The Temperature metric monitors the hotness or coldness of the temperature probe.

This metric is available only on Dell Poweredge Linux Systems.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The following table lists the metrics, their descriptions, and user actions.

Table 2–59 Temperature Metrics

Metric	Description	Data Source (SNMP MIB Object)
Current Temperature	Current reading of the temperature probe. The value is representing temperature in tenths of degrees Centigrade	temperatureProbeReading (1.3.6.1.4.1.674.10892.1.700.20.1.6)
Location	Description of the location name of the temperature probe. Examples of values are: "CPU Temp" and "System Temp".	temperatureProbeLocationName (1.3.6.1.4.1.674.10892.1.700.20.1.8)
Temperature Probe Status	See Section 2.34.1, "Temperature Probe Status"	See Section 2.34.1, "Temperature Probe Status"

2.34.1 Temperature Probe Status

Represents the status of the temperature probe.

This metric is available only on Dell Poweredge Linux Systems.

The following table lists the possible values for this metric and their meaning.

Metric Value	Meaning (per SNMP MIB)
1	Other (not one of the following)
2	Unknown
3	Normal
4	Warning
5	Critical
6	Non-Recoverable

Metric Summary

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

Table 2–60 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	Not Uploaded	>=	4	5	1	Temperature at probe %ProbeIndex% in chassis %ChassisIndex% is %TemperatureReading% (C). Status is %value%, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Chassis Index" and "Temperature Probe Index" objects.

If warning or critical threshold values are currently set for any unique combination of "Chassis Index" and "Temperature Probe Index" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Chassis Index" and "Temperature Probe Index" objects, use the Edit Thresholds page.

Data Source

SNMP MIB object: temperatureProbeStatus (1.3.6.1.4.1.674.10892.1.700.20.1.5)

2.35 Top Processes

The Top Processes metric is a listing of (up to) 20 processes that include 10 processes consuming the largest percentage of memory and 10 processes consuming the most percentage of CPU time. The processes are listed in the order of memory consumption.

The data sources for this metric category include the following:

Host	Data Source
Solaris	ps command
HP	ps command
Linux	ps command
HP Tru64	ps command
IBM AIX	ps command
Windows	performance data counters

The following table lists the metrics and descriptions.

Table 2–61 Top Processes Metrics

Metric	Description
Command and Arguments	Command and all its arguments
CPU Time for Top Processes	CPU utilization time in seconds
CPU Utilization for Top Processes (%)	Percentage of CPU time consumed by the process. For UNIX-based platforms, check the load on the system using the UNIX uptime or top commands. Also, check for processes using too much CPU time by using the top and ps -ef commands. Note that the issue may be a large number of instances of one or more processes, rather than a few processes each taking up a large amount of CPU time. Kill processes using excessive CPU time.
Memory Utilization for Top Processes (%)	Percentage of memory consumed by the process
Physical Memory Utilization (KB)	Number of kilobytes of physical memory being used. For Solaris and IBM AIX hosts, the data source is kernel memory structure (struct vminfo).
Process User ID	User name that owns the process, that is, the user ID of the process being reported on. For the Windows host, the data source is the Windows API.
Virtual Memory Utilization (KB)	Total size of the process in virtual memory in kilobytes (KB). For the Windows host, the data source is the Windows API.

2.36 TTY Activity

This metric reports tty device activity.

The data sources for this metric include the following:

Host	Data Source
Solaris	sar command
HP	sar command
Linux	not available
HP Tru64	table() system call
IBM AIX	sar command
Windows	not available

The OS sar command is used to sample cumulative activity counters maintained by the OS. The data is obtained by sampling system counters once in a five-second interval.

The following tables lists the metrics and their descriptions.

Table 2–62 TTY Activity Metrics

Metric	Description
Incoming Character Interrupts (per second)	Number of received incoming character interrupts per second
Input Characters Processed by canon()	Input characters processed by canon() per second
Modem Interrupt Rate (per second)	Modem interrupt rate
Outgoing Character Interrupts (per second)	Number of transmit outgoing character interrupts per second
TTY Output Characters (per second)	Number of output characters per second
TTY Raw Input (chars/s)	Raw input characters per second

2.37 User Defined Metrics

The UDM metric allows you to execute your own scripts. The data returned by these scripts can be compared against thresholds and generate severity alerts similar to alerts in predefined metrics. UDM is similar to the Oracle9*i* Management Agent's UDE functionality.

The data source for these metrics is the User Defined Script.

The following table lists the metrics and their descriptions.

Table 2–63 User Defined Metrics

Metric	Description
User Defined Numeric Metric	Contains a value if the value type is NUMBER. Otherwise, the value is "", if the value is STRING.
User Defined String Metric	Contains a value if the value type is STRING. Otherwise, the value is "", if the value is NUMBER.

2.38 Users

The Users metric provides information about the users currently on the system being monitored.

2.38.1 Number of Logons

Represents the number of times a user with a certain user name is logged on to the host target.

Data Source

For Solaris, HP, Linux, HP Tru64, and IBM AIX, the number of times a user is logged on is obtained from the OS w command.

For Windows, the source of information is Windows API.

2.39 Windows Events Log

The purpose of this metric is to collect those entries from all available Windows NT event log files whose type is either Error or Warning. A critical or a warning alert is raised only for System and Security Event log file entries.

Note: Since log files continue to grow, this metric outputs log events which had been written to the log file after the last collection time, that is, only those records are written out whose timeGenerated (time when the event was generated) is after the last collection time until the last record of the log file. If this metric is collected for the first time, only the events generated on the *current date* are outputted.

This metric is available only on Windows.

Note: For all target versions, the collection frequency for each metric is every 15 minutes.

The data source for these metrics is WMI Operating System Classes.

The following table lists the metrics and their descriptions.

Table 2–64 Windows Events Log Metrics

Metric	Description
Category	Subcategory for this event. This subcategory is source-specific.
Date-Time	Date and time when the Source generated the event.
Description	Event message as it appears in the Windows event log.
Event ID	Identifier of the event
Log Name	Name of the Windows event log file
Record Number	Identifies the event within the Windows event log file
Source	Name of the source (application, service, driver, subsystem) that generated the entry
User	Name of the logged-on user when the event occurred. If the user name cannot be determined, the user name is NULL.
Windows Event Severity	See Section 2.39.1, "Windows Event Severity"

2.39.1 Windows Event Severity

The seriousness of the event. Possible values are: Warning and Error.

This metric is available only on Windows.

Metric Summary

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

Table 2–65 Metric Summary Table

Target Version	Key	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	logfile: "system"	Every 15 Minutes	After Every Sample	=	warning	error	1*	X1User[%user%]:Category[%categorystring%]:Description[%message%]

* Once an alert is triggered for this metric, it must be manually cleared.

Multiple Thresholds

For this metric you can set different warning and critical threshold values for each unique combination of "Log Name", "Source", and "Event ID" objects.

If warning or critical threshold values are currently set for any unique combination of "Log Name", "Source", and "Event ID" objects, those thresholds can be viewed on the Metric Detail page for this metric.

To specify or change warning or critical threshold values for each unique combination of "Log Name", "Source", and "Event ID" objects, use the Edit Thresholds page.

Data Source

WMI Operating System Classes

2.40 Zombie Processes

The Zombie Processes metric monitors the orphaned processes in the different variations of UNIX systems.

2.40.1 Processes in Zombie State (%)

Represents the percentage of all processes running on the system that are currently in zombie state.

Metric Summary

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

Table 2–66 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 15 Minutes	After Every 60 Samples	>	35	50	1	%value%% of all processes are in zombie state, crossed warning (%warning_threshold%) or critical (%critical_threshold%) threshold.

Data Source

The data sources for this metric include the following:

Host	Data Source
Solaris	ps command
HP	ps command
Linux	ps command
HP Tru64	not available
IBM AIX	not available
Windows	not available

3

OMS and Repository

The OMS and Repository target exposes metrics that are useful for monitoring the Oracle Enterprise Manager Management Service (OMS) and Management Repository.

3.1 Active Loader Status

This category of metrics provides information on Active Loader Status per OMS.

3.2 Active Management Servlets

This category of metrics provides information on Active Management Servlets Category.

3.3 Agent Status

This category of metrics provides information on the agent status.

3.4 Cleared Group Security Violations

This category of metrics provides information on the violations on cleared group security.

3.5 Cleared Target Security Violations

This category of metrics provides information on the violations on cleared target security.

3.6 Configuration

This category of metrics provides information on configuration.

3.7 DBMS Job Status

This category of metrics provides information on the DBMS job status.

3.8 Duplicate Targets

This category of metrics provides information on duplicate targets.

3.9 Job Dispatcher Performance

This category of metrics provides information on the performance of job dispatcher.

3.10 New Group Security Violations

This category of metrics provides information on the security violations on new groups.

3.11 New Target Security Violations

This category of metrics provides information on the security violations on new targets.

3.12 No Agents

This category of metrics provides information on no agents.

3.13 Notification Method Performance

This category of metrics provides information on the performance of notification methods.

3.14 Notification Performance

This category of metrics provides information on the performance of notifications.

3.15 Notification Status

This is a Management Agent metric intended to send out of band notifications when the Notification system is determined to be in a critical state.

3.15.1 Average Delivery Time (ms)

This metric should be used in conjunction with Notifications Waiting for the particular notification type to help determine if a notification problem is becoming worse. If the notifications waiting is increasing along with the average delivery time, then the problem is most likely in the delivery, not the number of notifications itself. Delivery time problems can be related to network problems or resource constraints.

Data Source

The data for this metric comes from entries in the mgmt_system_performance_log where name=<method_name>||_TOTAL_DELIVERY_TIME

User Action

If the value is steadily increasing perform the following user actions:

1. Check the Errors page for errors logged by Notification Delivery.
2. Check for resource constraints along the notification delivery path e.g. network errors, email or snmp servers being down etc.

3.15.2 Cleared Group Security Violations

This metric collects the information about the cleared violations on all groups of targets having security policies defined for the member targets. The number of cleared violations will increase with more violations getting rectified. This is used to trend the rate of fix of security policy violations.

Data Source

The data for this metric comes from entries in the mgmt_policies, mgmt_violations and mgmt_flat_target_assoc.

User Action

If the number of cleared violations is static or there are no cleared violations, check the security policy violations and ensure that the violations are rectified as recommended by the policy.

3.15.3 Cleared Target Security Violations

This metric collects the information about the cleared violations on all targets having security policies defined for them. The number of cleared violations will increase with more violations getting rectified. This is used to trend the rate of fix of security policy violations.

Data Source

The data for this metric comes from entries in the mgmt_policies, mgmt_violations and mgmt_flat_target_assoc.

User Action

If the number of cleared violations is static or there are no cleared violations, check the security policy violations and ensure that the violations are rectified as recommended by the policy.

3.15.4 DBMS Job Bad Schedule

This metric flags a DBMS job whose schedule is invalid. A schedule is marked 'Invalid' if it is scheduled for more than one hour in the past, or more than one year in the future. An invalid schedule means that the job is in serious trouble.

Data Source

The user_jobs.next_time table in the Management Repository.

User Action

If the job schedule is invalid, the DBMS job should be restarted. To do this:

1. Copy down the DBMS Job Name that is down from the row in the table. This DBMS Job Name is 'yourDBMSjobname' in the following example.
2. Logon to the database as the repository owner.
3. Issue the following SQL statement:

```
select dbms_jobname
  from mgmt_performance_names
  where display_name='yourDBMSjobname';
```

4. If the dbms_jobname is 'myjob', then issue the following SQL statement:

```
select job
  from all_jobs
  where what='myjob' ;
```

5. Copy down the jobid.
6. Force the job into the broken state so that it can be restarted by specifying the following DBMS job command and parameters:

   `dbms_job.broken(jobid,true)`

7. Verify that the job has been marked as broken by using this SQL statement:

```
select what, broken
  from all_jobs
  where broken='Y';
```

   You should see the job in the results.

8. Once you've verified that the DBMS job is marked broken, restart the job with the following DBMS job command and parameters:

   `dbms_job.run(jobid)`

3.15.5 DBMS Job Processing Time, % of Last Hour

The percentage of the past hour the job has been running.

Data Source

The mgmt_system_performance_log table in the Management Repository.

User Action

If the value of this metric is greater than 50%, then there may be a problem with the job. Check the System Errors page for errors reported by the job. Check the Alerts log for any alerts related to the job.

3.15.6 DBMS Job UpDown

The down condition equates to the dbms_job "broken" state. The Up Arrow means not broken.

Data Source

The broken column is from the all_users table in the Management Repository.

User Action

Determine the reason for the dbms job failure. Once the reason for the failure has been determined and corrected, the job can be restarted through the dbms_job.run command.

To determine the reason the dbms job failed, take the following steps (replacing myjob with the displayed name of the down job):

1. Copy down the DBMS Job Name that is down from the row in the table. This DBMS Job Name is 'yourDBMSjobname' in the following example.
2. Log onto the database as the repository owner.
3. Issue the following SQL statement:

```
select dbms_jobname
```

```
  from mgmt_performance_names
  where display_name='yourDBMSjobname';
```

4. If the dbms_jobname is 'myjob', then issue the following SQL statement:

```
select job
  from all_jobs
  where what='myjob';
```

5. Using the job id returned, look for ORA-12012 messages for this jobid in the alerts log and trace files and try to determine and correct the problem.

The job can be manually restarted through the following database command:

```
execute dbms_job.run (jobid);
```

3.15.7 Files Pending Load

The number of files waiting for the loader to process, sampled every 10 minutes.

Data Source

This metric is obtained using the following query of the mgmt_oms_parameters table in the Management Repository.

```
SELECT value
  FROM mgmt_oms_parameters
  where name='loaderFileCount'
```

User Action

If the Files Pending Load number is increasing steadily over a period of time, you may consider one of these options:

- Increasing the number of background threads.
- Adding another Management Service and pointing some of the Management Agents to the new Management Service.

3.15.8 Group Compliance

This metric gives the average of compliance score of policy rules associated with its member targets and self-target itself. The compliance score ranges from 0-100 %. This metric is collected for every 6 hours (360 minutes).

It tells how well the group is compliant with policy rules.

Data Source

The data for the metric comes from entries in the MGMT_POLICY_ASSOC_EVAL_ SUMM.

User Action

If the value increases steadily, perform the following:

1. Check the groups policy rule data in Policy Violations tab and check for the individual compliance score of the policy rules of the member targets and self-target.
2. Concentrate on policy rules, which have lesser compliance score and try to resolve the corresponding policy rule violations manually or through automatic corrective actions.

3.15.9 Group Security Compliance

This metric is used to collect the compliance trend of all the groups of targets w.r.t the security policies defined on the member targets. The security compliance score is an indication of the security health of a target. A score of 100 indicates full compliance and a score of 0 indicates no compliance.

Data Source

The data for this metric comes from entries in the mgmt_policy_assoc_eval_summ.

User Action

If the compliance score is reducing continuously, check the security policy violations and ensure that the violations are rectified as recommended by the policy.

3.15.10 Group Target Compliance

This metric gives the average of the compliance score of all policy rules associated with each of its member targets and self-target. The metric data is rolled up by each member target so that the user can get the member target-wise compliance score of a group.

It helps to show the trend data on how many targets of the group lies in good compliance score range and how many are in poor compliance score range.

The metric is collected for every 6 hours (360 minutes).

Data Source

The compliance score of policies evaluated is in the mgmt_policy_assoc_eval_summ table.

User Action

If the average compliance score is coming down check the security policy violations in the Security At a Glance page. Identify the violating policies and fix the violation. Details of the violations and their policies can be had from the Policy Violations page.

3.15.11 Group Violations

This metric gives the sum of the violations of all policy rules associated with member targets of the group and self-target. Along with the violations count, it has the violation level also to tell whether it is Critical/Warning/Informational violation. It helps to show the trend overview of group policy violations data. This metric is collected for every 6 hours (360 minutes).

Data Source

The data for the metric comes from entries in the MGMT_POLICY_ASSOC_EVAL_ SUMM.

User Action

If the value increases steadily, perform the following:

1. Check the policy violations of the group target and its member targets.
2. Give more priority to Critical violations, then warning and informational. Check the policy rules causing the policy violations in policy violations tab page.

3. Try to resolve the violations through automatic corrective actions or manual actions.

3.15.12 Job Dispatcher Job Step Average Backlog

The number of job steps that were ready to be scheduled but could not be because all the dispatchers were busy.

When this number grows steadily, it means the job scheduler is not able to keep up with the workload.

User Action

This is the sum of job steps whose next scheduled time is in the past - job steps eligible to run but not yet running. If the graph of this number increases steadily over time, the user should take one of the following actions:

- Increase the `em.jobs.shortPoolSize`, `em.jobs.longPoolSize` and `em.jobs.systemPoolSize` properties in the web.xml file. The web.xml file specifies the number of threads allocated to process different types of job steps. The short pool size should be larger than the long pool size.

Property	Default Value	Recommended Value	Description
em.jobs.shortPoolSize	10	10 50	Steps taking less than 15 minutes
em.jobs.longPoolSize	8	8 - 30	Stars taking more than 15 minutes
em.jobs.systemPoolSize	8	8 - 20	Internal jobs (e.g. agent ping)

- Add another Management Service on a different host.

Check the job step contents to see if they can be made more efficient.

3.15.13 Job Dispatcher Processing Time, % of Last Hour

The job dispatcher is responsible for scheduling jobs as required. It starts up periodically and checks if jobs need to be run. If job dispatcher is running more than the threshold levels, then it is having problems handling the job load.

Data Source

This is the sum of the amount of time the job has run over the last hour from the mgmt_system_performance_log table in the Management Repository divided by one hour, multiplied by 100 to arrive at the percent.

3.15.14 Last Error

Timestamp of the latest error for the job.

Data Source

The mgmt_system_error_log table in the Management Repository.

3.15.15 Loader Directory

The directory from which the loader is getting files.

Data Source

This metric is obtained using the following query of the mgmt_oms_parameters table in the Management Repository.

```
SELECT value
  FROM mgmt_oms_parameters
  where name='loaderDirectory'
```

User Action

If the loader directory is out of space, you may want to look for the error files to investigate the problem.

3.15.16 Loader Name

The unique name of the loader, consisting of the Management Service name separated by a comma from the loader name on that Management Service.

Data Source

The mgmt_system_performance_log table in the Management Repository.

3.15.17 Loader Throughput (rows per hour)

This is the number of lines of XML text processed by the loader thread over the past hour.

Data Source

The mgmt_system_performance_log table in the Management Repository.

User Action

If this number continues to rise over time, then the user may want to consider adding another Management Service or increasing the number of loader threads for this Management Service. To increase the number of loader threads, add or change the `em.loader.threadPoolSize` entry in the emoms.properties file. The default number of threads is 2. Values between 2 and 10 are common.

3.15.18 Loader Throughput (rows per second)

This is the number of lines of XML text processed by the loader thread per second averaged over the past hour.

Data Source

The mgmt_system_performance_log table in the Management Repository.

3.15.19 Management Service Status

Shows whether the Management Service is up or down.

Data Source

The mgmt_oms_parameters and mgmt_failover_table tables in the Management Repository.

User Action

If the Management Service is down, start it. Only management services that are down can be deleted.

3.15.20 New Group Security Violations

This metric collects the information about the new violations that have happened on all groups of targets having security policies defined for the member targets. The number of new violations will increase with newer violations and will decrease with the violations getting cleared. This is used to trend the rate of arrival of new security policy violations.

Data Source

The data for this metric comes from entries in the mgmt_policies, mgmt_violations.

User Action

If the number of new violations is increasing continuously, check the security policy violations and ensure that the violations are rectified as recommended by the policy.

3.15.21 New Target Security Violations

This metric collects the information about the new violations that have happened on all targets having security policies defined for them. The number of new violations will increase with newer violations and will decrease with the violations getting cleared. This is used to trend the rate of arrival of new security policy violations.

Data Source

The data for this metric comes from entries in the mgmt_policies, mgmt_violations.

User Action

If the number of new violations is increasing continuously, check the security policy violations and ensure that the violations are rectified as recommended by the policy.

3.15.22 Notification Delivery Time

The time it took to deliver a notification, averaged over the past hour.

Data Source

The mgmt_system_performance_log table in the Management Repository.

User Action

If the average delivery time is steadily increasing, verify that the notification methods specified are valid. Remove any unnecessary or out of date notification rules and schedules.

3.15.23 Notification Processing Time, % of Last Hour

The percentage of the past hour that Notification delivery has been running.

Data Source

The mgmt_system_performance_log table in the Management Repository.

User Action

If the average delivery time is steadily increasing, verify that the notification methods specified are valid. Remove any unnecessary or out of date notification rules and schedules.

3.15.24 Notification UpDown

Displays whether the notification DBMS job (which processes severities to determine if notifications are required) is up or down.

Data Source

The user_jobs table in the Management Repository.

User Action

Determine the reason for the DBMS job failure. Once the reason for the failure has been determined and corrected, the job can be restarted through the dbms_job.run command.

To determine why the DBMS job failed, perform the following steps:

1. Logon to the database as the Management Repository owner.
2. Issue the following SQL statement:

```
select job
  from all_jobs
  where what like '%CHECK_FOR_SEVERITIES%';
```

3. Using the job id returned, look for ORA-12012 messages for this jobid in the alerts log and trace files and try to determine and correct the problem.
4. Issue the following DBMS job command and parameters:

```
execute dbms_job.run (jobid);
```

3.15.25 Notifications Processed

The total number of notifications delivered by the Management Service over the previous 10 minutes.

Data Source

The mgmt_system_performance_log table in the Management Repository.

User Action

If the number of notifications processed is continually increasing over several days, then you may want to consider adding another Management Service.

3.15.26 Notifications Waiting

Notification Method Performance metrics measure the performance data for each notification type, such as SNMP, EMAIL, OSCMD, PLSQL and RCA. This metric shows the number of notifications queued for the method type.

Data Source

The data for this metric comes from entries in the mgmt_system_performance_log where name=<method_name>||_S_QUEUED

User Action

If the value is steadily increasing perform the following user actions:

1. Check the Errors page for errors logged by the Notification Delivery.

2. Check the number of notification rules defined utilizing the method and verify that they are all necessary, removing those that are not.
3. Verify that the addresses being used for the notifications are correct

3.15.27 Number of Active Agents

The number of active agents in the repository. If this number is 0, then Enterprise Manager is not monitoring any external targets. May be a problem if unexpected.

Data Source

The number of agents whose status is up in the mgmt_current_availability table.

User Action

If no agents are running, determine the reasons they are down, correct if needed and restart. Log files in the agent's $ORACLE_HOME/sysman/log directory can provide information about possible causes of agent shutdown.

3.15.28 Number of Administrators

The number of administrators defined for Enterprise Manager.

Data Source

The mgmt_created_users table in the Management Repository.

3.15.29 Number of Duplicate Targets

The count of duplicate targets in the Management Repository.

Data Source

The mgmt_duplicate_targets table in the Management Repository.

User Action

Go to the Duplicate Targets page by clicking the **Duplicate targets** link on the Management System Overview page. The **Duplicate targets** link only appears on the Management System Overview page if there are problems involving duplicate targets.

Resolve the conflict by removing the duplicate target from the conflicting Management Agent.

3.15.30 Number of Groups

The number of groups defined for Enterprise Manager.

Data Source

The mgmt_targets table in the Management Repository.

User Action

If you have a problem viewing the All Targets page, you may want to check the number of roles and groups.

3.15.31 Number of Roles

The number of roles defined for Enterprise Manager.

Data Source

The mgmt_roles table in the Management Repository.

User Action

If you have a problem viewing the All Targets page, you may want to check the number of roles and groups.

3.15.32 Number of Targets

The number of targets defined for Enterprise Manager.

Data Source

The mgmt_targets table in the Management Repository.

3.15.33 Oldest Loader File

This metric shows how long the oldest loader file has been waiting to be processed by the loader. This is an indicator of the delay from when the Management Agent sends out information to when the user receives the information.

Data Source

This metric is obtained using the following query of the mgmt_oms_parameters table in the Management Repository.

```
SELECT value
  FROM mgmt_oms_parameters
  where name='loaderOldestFile'
```

User Action

If the oldest loader file is extremely old, you have a loader problem. You may want to add another Management Service and point some of the Management Agents to the new Management Service.

3.15.34 Repository Tablespace Used

This is the total number of MB that the Management Repository tablespaces are currently using.

Data Source

The dba_data_files table in the Management Repository.

3.15.35 Restart Count

The number of times the agent has been restarted in the past 24 hrs.

Data Source

Derived by:

```
(SELECT t.target_name, COUNT(*) down_count
  FROM mgmt_availability a, mgmt_targets t
  WHERE a.start_collection_timestamp = a.end_collection_timestamp
    AND a.target_guid = t.target_guid
    AND t.target_type = MGMT_GLOBAL.G_AGENT_TARGET_TYPE
    AND a.start_collection_timestamp > SYSDATE-1
  GROUP BY t.target_name)
```

User Action

If this number is high, check the agent logs to see if a system condition exists causing the system to bounce. If an agent is constantly restarting, the Targets Not Uploading Data metric may also be set for targets on the agents with restart problems. Restart problems may be due to system resource constraints or configuration problems.

3.15.36 Session Count

A count of the number of sessions between the Management Service and Management Repository database.

Data Source

The gv$session system view.

3.15.37 Steps Per Second

The number of job steps processed per second by the job dispatcher, averaged over the past hour and sampled every 10 minutes.

Data Source

The mgmt_job_execution table in the Management Repository.

3.15.38 Target Addition Rate (Last Hour)

The rate at which targets are being created. The target addition rate should be greatest shortly after EM is installed and then should increase briefly whenever a new agent is added. If the rate is increasing abnormally, you should check for abnormal agent or administrator activity and verify that the targets are useful. Check to see that group creation is not being over utilized.

Data Source

The metric is derived from the mgmt_target table, the current target count - target count at last sampling.

3.15.39 Target Compliance

This metric gives the compliance score for each target. It is calculated based on the compliance score of the individual policy rules associated with the given target. Compliance score ranges from 0-100 and it is represented in percentage. It tells how good the target is complaint with associated policy rules. This metric is collected for every 6 hours (360 minutes).

Data Source

The data for the metric comes from entries in the MGMT_POLICY_ASSOC_EVAL_SUMM

User Action

If the value decreases steadily, perform the following:

1. Check the targets policy rule data in Policy Violations tab and check for the individual compliance score of the policy rules of the target.
2. Concentrate on policy rules, which have lesser compliance score and try to resolve the corresponding policy rule violations manually or through automatic corrective actions.

3.15.40 Target Security Compliance

This metric is used to capture the compliance trend of all the targets w.r.t the security policies defined on them. The security compliance score is an indication of the security health of a target. A score of 100 indicates full compliance and a score of 0 indicates no compliance.

Data Source

The data for this metric comes from entries in the mgmt_policy_assoc_eval_summ.

User Action

If the compliance score is reducing continuously, check the security policy violations and ensure that the violations are rectified as recommended by the policy.

3.15.41 Target Violations

This metric gives the sum of the violations of all policy rules associated with each target. Along with the violations count, it has the violation level also to tell whether it is Critical/Warning/Informational violation. This metric is collected for every 6 hours (360 minutes).

It helps to show the trend overview of target policy violations data.

Data Source

The data for the metric comes from entries in the MGMT_POLICY_ASSOC_EVAL_SUMM.

User Action

If the value increases steadily, perform the following:

1. Give more priority to Critical violations, then warning and informational. Check the policy rules causing the policy violations in policy violations tab page.
2. Try to resolve the violations through automatic corrective actions or manual actions.

3.15.42 Throughput Per Second

The number of notifications delivered per second, averaged over the past hour.

Data Source

The mgmt_system_performance_log table in the Management Repository.

3.15.43 Total Loader Runtime in the Last Hour (seconds)

This is the amount of time in seconds that the loader thread has been running in the past hour.

Data Source

The mgmt_system_performance_log table in the Management Repository.

User Action

If this number is steadily increasing along with the Loader Throughput (rows per hour) metric, then perform the actions described in the User Action section of the help topic for the Loader Throughput (rows per hour) metric. If this number increases but

the loader throughput does not, check for resource constraints, such as high CPU utilization by some process, deadlocks in the Management Repository database, or processor memory problems.

3.15.44 Total Repository Tablespace

The total MB allocated to the Management Repository tablespaces. This will always be greater than or equal to the space used.

Data Source

The dba_free_space table in the Management Repository.

3.15.45 User Addition Rate (Last Hour)

The rate at which users are being created. The target addition rate should be low. If the rate is increasing abnormally, you should check for abnormal administrator activity.

Data Source

The metric is derived from the mgmt_created_users table, the current user count - user count at last sampling.

3.16 Oracle Management Services and Repository

The OMS and Repository target exposes metrics that are useful for monitoring the Oracle Enterprise Manager Management Service (OMS) and Management Repository.

3.17 Repository Collections Performance

This category of metrics provides information on the performance of repository collections. They are collected by background dbms jobs in the repository database called collection workers. Repository metrics are sub divided into long and short running metrics. These are called task classes (short task class and long task class). Some collection workers (Default 1) process the short task class and some (Default 1) process long task class. Repository collection performance metrics measure the performance data for repository metric collections for each task class. This metric is a repository metric and hence collected by the collection workers.

3.18 Repository Job Dispatcher

This category of metrics provides information on the Repository Job Dispatcher.

3.18.1 Collection Duration (seconds)

The total amount of time in seconds the collection workers were running in last 10 minutes. This is an indicator of the load on the repository collection subsystem. This could be due to two reasons, the number of collections have increased or some of the metrics are taking a long time to complete. This needs to be related with collections processed metric to find out if number of collections have increased or metrics are taking a long time.

Data Source

The data for this metrics come from entries in mgmt_system_performance log where job_name=MGMT_COLLECTION.Collection Subsystem.

3.18.2 Collections Processed

The total number of collections that were processed in the last 10 minutes.

Data Source

The data for this metrics come from entries in mgmt_system_performance log where job_name=MGMT_COLLECTION.Collection Subsystem

3.18.3 Collections Waiting To Run

The total number of collections that were waiting to run at the point this metric was collected. An increasing value would mean the collection workers are falling behind and would need to be increased. The collections waiting to run could be high initially on system startup and should ideally go down towards zero.

Data Source

The data for this metrics come from entries in mgmt_collection_tasks table which holds all the list of collections.

3.18.4 Number of Workers

The total number of workers that were processing the collections.

Data Source

The data for this metric come from entries in mgmt_collection_workers table.

3.18.5 Total Throughput Across Workers

The total number of collections per second processed by all the collection workers.

Data Source

The data for this metrics come from entries in mgmt_system_performance log where job_name=MGMT_COLLECTION.Collection Subsystem.

3.19 Repository Sessions

This category of metrics provides information on the Repository sessions.

3.20 Response

This page indicates whether Enterprise Manager is up or down. It contains historical information for periods in which it was down.

3.20.1 Status

This metric indicates whether Enterprise Manager is up or down. If you have configured the agent monitoring the oracle_emrep target with a valid email address, you will receive an email notification when Enterprise Manager is down.

Metric Summary

The following table shows how often the metric's value is collected and compared against the default thresholds. The 'Consecutive Number of Occurrences Preceding

Notification' column indicates the consecutive number of times the comparison against thresholds should hold TRUE before an alert is generated.

Table 3–1 Metric Summary Table

Target Version	Evaluation and Collection Frequency	Upload Frequency	Operator	Default Warning Threshold	Default Critical Threshold	Consecutive Number of Occurrences Preceding Notification	Alert Text
All Versions	Every 5 Minutes	Not Uploaded	=	Not Defined	0	1	%Message%

Data Source

sysman/admin/scripts/emrepresp.pl

User Action

This metric checks for the following:

- Is the Management Repository database up and accessible?

  If the Management Repository database is down, start it. If 'Invalid Username or Password' error is displayed, verify that the name and password for the oracle_emrep target is the same as the repository owner's name and password.

- Is at least one Management Service running?

  If a Management Service is not running, start one.

- Is the Repository Metrics dbms job running?

  If the DBMS job is down or has an invalid schedule, it should be restarted by following the instructions in the User Action section of the help topic for the DBMS Job Bad Schedule metric.

4

Services Metrics

This chapter describes the Services Metrics.

4.1 DNS Response Metrics

The following sections lists the DNS Response metrics, their descriptions, and user actions.

User Action

If TTL values are low, then you may consider configuring your DNS service to allow caching for longer periods of time.

4.1.1 [DNS] Number of Results

A response to a DNS request may contain multiple answers. This indicates the number of answers (e.g. the number of IP addresses) in the response from the DNS service.

User Action

If the number of results is not what it should be, then you will need to examine your DNS service's configuration.

4.1.2 [DNS] Status

Indicates whether the query was successful.

User Action

Consult the Results metric for details.

4.1.3 [DNS] Total Connect Time (ms)

Time taken to connect to the DNS service. This metric is collected for queries using the TCP protocol.

User Action

A slow Total Connect Time suggests that network congestion is a problem.

4.1.4 [DNS] Total Response Time (ms)

Total time required to receive a response from the DNS service.

User Action

Unusually slow response times can occur if the DNS server has to perform a lookup before it can respond. If the total response time is consistently slow, then either the network may be slow, or the DNS server may be having trouble generating a response. Try using traceroute to diagnose network issues

4.1.5 [DNS] TTL (seconds)

A response to a DNS request may contain multiple answers. Each answer in a DNS response has a TTL (Time To Live) that indicates the number seconds that the answer may be cached by a client. The TTL value reported here is the minimum TTL of all the answers in the DNS response.

User Action

If TTL values are low, then you may consider configuring your DNS service to allow caching for longer periods of time.

4.1.6 DNS Results

The results of the DNS query. In the event of a status down, the reason for the status down will be displayed here.

4.2 FTP Response Metrics

The following table lists the FTP Response metrics and their descriptions.

Table 4–1 FTP Response Metrics

Metric	Description
[FTP] Connect Time (ms)	Time taken to establish a connection with the FTP service.
[FTP] Download Rate (KB/second)	Rate at which the data is downloaded from the FTP service.
[FTP] Download Time (ms)	Total time taken to download a file from the service
[FTP] Login Time (ms)	Time required to login to the FTP service.
[FTP] NOOP Time (ms)	Time required to perform a NOOP. A NOOP ("No Operation") is a request that the FTP service respond with an "OK" status. A NOOP is similar to network round-trip time because generating a response to a NOOP requires minimal effort.
[FTP] Number of Retries	Number of retries required before the test was successful.
[FTP] Status	Indicates that all requests succeeded and that the downloaded file matched the uploaded file.
[FTP] Status Description	Details on the reason for any service failure.
[FTP] Total Time (ms)	The total time required to test the FTP service.
[FTP] Upload Time (ms)	Time taken to upload a file to the service.
[FTP] Upload Rate	Rate at which the data is downloaded from the FTP service.

4.3 HTTP Raw Metrics

The following sections lists the HTTP Raw metrics, their descriptions, and user actions.

4.3.1 HTTP Raw Time Per Connection

This metric measures the average connect time for all pages in the transaction. This is calculated as: Total Connect Time / Number of Connections Made. The Connect Time is one of the phases of a transaction that can help you isolate and fix response time problems.

User Action

The average connect time, when reviewed over a period of time, can indicate whether network congestion or other connectivity issues are the cause of poor Web application response time.

4.3.2 HTTP Raw Broken URL Count

This metric measures the number of errors encountered when displaying content for the pages accessed by the transaction, step or step group. For example, missing GIF images or style sheets will increase the value of the Broken Count metric.

User Action

Use this metric to measure the quality of the pages being served by your Web application. For example, high values for the Broken Count metric can indicate that files have been moved or that relative links in the application are broken.

4.3.3 HTTP Raw Broken URL Details

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

4.3.4 HTTP Raw Connect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction, step or step group can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spent connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

4.3.5 HTTP Raw First Byte Time (ms)

This is the First Byte Time divided by the number of pages in the step, step group, or transaction.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.3.6 HTTP Transaction DNS Time

This metric is not supported for this version of Enterprise Manager.

4.3.7 HTTP Raw HTML Time (ms)

Enterprise Manager breaks down each step, step group, or transaction into individual phases. Performance metrics for each phase can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

4.3.8 HTTP Raw Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

4.3.9 HTTP Raw Perceived Slowest Page / Page Element Time (ms)

The amount of time that it would take a Web browser to play the slowest page in the step, step group, or transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that's slow to response respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.3.10 HTTP Raw Perceived Time per Page / Page Element (ms)

The average amount of time that it would take a Web browser to play each page in the step, step group, or transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.\

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.3.11 HTTP Raw Perceived Total Time (ms)

Indicates the overall time spent to process the step, step group, or transaction. This includes all the phases of the step / step group / transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.3.12 HTTP Raw Redirect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a transaction. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

4.3.13 HTTP Raw Status

Indicates whether the Web transaction was successful.

4.3.14 HTTP Raw Status Description

If the beacon is unable to run the step, step group, or transaction successfully, this metric returns a description of the error.

4.3.15 HTTP Raw Total Time (ms)

Indicates the overall time spent to process the step, step group, or transaction. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that is slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.3.16 HTTP Raw Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

4.3.17 HTTP Raw First Byte Time

Enterprise Manager breaks down each transaction, step or step group into individual phases. Performance metrics for each phase can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.3.18 HTTP Raw URL

This is the URL associated with the step.

4.4 HTTP Step Group Metrics

The following sections lists the HTTP Step Group metrics, their descriptions, and user actions.

4.4.1 [HTTP Step Group] Connect Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the step group can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spend connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

4.4.2 [HTTP Step Group] Broken URL Count

This metric measures the number of errors encountered when displaying content for the pages accessed by the step group. For example, missing GIF images or style sheets will increase the value of the Broken Count metric.

User Action

Use this metric to measure the quality of the pages being served by your Web application. For example, high values for the Broken Count metric can indicate that files have been moved or that relative links in the application are broken.

4.4.3 [HTTP Step Group] First Byte Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.4.4 [HTTP Step Group] Broken URL Details

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

4.4.5 [HTTP Step Group] First Byte Time per Page (ms)

This is the First Byte Time divided by the number of pages in the step group.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.4.6 [HTTP Step Group] HTML Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

4.4.7 [HTTP Step Group] DNS Time

This metric is not supported for this version of Enterprise Manager.

4.4.8 [HTTP Step Group] Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

4.4.9 [HTTP Step Group] Perceived Slowest Page Time (ms)

The amount of time that it would take a Web browser to play the slowest page in a step group. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that's slow to response respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem

4.4.10 [HTTP Step Group] Perceived Time per Page (ms)

The average amount of time that it would take a Web browser to play each page in the step group. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the step group to isolate and repair the problem.

4.4.11 [HTTP Step Group} Perceived Total Time (ms)

The amount of time that it would take a Web browser to play the step group. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem transactions. After you identify a step group that's slow to respond to user requests, you can drill down and analyze each phase of the step group to isolate and repair the problem.

4.4.12 [HTTP Step Group] Redirect Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the step group can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a step group. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

4.4.13 [HTTP Step Group] Status

Indicates whether the Web transaction was successful.

4.4.14 [HTTP Step Group] Status Description

If the beacon is unable to run the step group successfully, this metric returns a description of the error that prevented the step group from running.

4.4.15 [HTTP Step Group] Time per Connection (ms)

This is the Connect Time divided by the number of connections made while playing a step group.

User Action

Slow Time per Connection has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner. Transactions that use HTTPS will typically have a much higher Time per Connection than transactions that use HTTP.

4.4.16 [HTTP Step Group] Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

4.4.17 [HTTP Step Group] Total Time (ms)

Indicates the overall time spent in processing the step group. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.5 HTTP Transaction Metrics

The following sections lists the HTTP Transaction metrics, their descriptions, and user actions.

4.5.1 [HTTP Transaction] Connect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spend connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

4.5.2 [HTTP Transaction] First Byte Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.5.3 [HTTP Transaction] First Byte Time per Page (ms)

This is the First Byte Time divided by the number of pages in the transaction.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.5.4 [HTTP Transaction] Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

4.5.5 [HTTP Transaction] HTML Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

4.5.6 [HTTP Transaction] Perceived Slowest Page Time (ms)

The amount of time that it would take a Web browser to play the slowest page in the transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that is slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.5.7 [HTTP Transaction] Perceived Time per Page (ms)

The average amount of time that it would take a Web browser to play each page in the transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.5.8 [HTTP Transaction] Perceived Total Time

The amount of time that it would take a Web browser to play the transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.5.9 [HTTP Transaction] Redirect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a

transaction. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

4.5.10 [HTTP Transaction] Status

Indicates whether the Web transaction was successful.

4.5.11 [HTTP Transaction] Status Description

If the beacon is unable to run the transaction successfully, this metric returns a description of the error that prevented the transaction from running.

4.5.12 [HTTP Transaction] Time per Connection (ms)

This is the Connect Time divided by the number of connections made while playing a transaction.

User Action

Slow Time per Connection has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner. Transactions that use HTTPS will typically have a much higher Time per Connection than transactions that use HTTP.

4.5.13 [HTTP Transaction] Total Time (ms)

Indicates the overall time spent to process the transaction. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a r transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.5.14 [HTTP Transaction] Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

4.6 HTTP User Action Metrics

The following sections lists the HTTP User Action metrics, their descriptions, and user actions.

4.6.1 [HTTP Step] Connect Time (ms)

Enterprise Manager breaks down each step into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spend connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

4.6.2 [HTTP Step] Broken URL Count

This metric measures the number of errors encountered when displaying content for the pages accessed by the step. For example, missing GIF images or style sheets will increase the value of the Broken Count metric.

User Action

Use this metric to measure the quality of the pages being served by your Web application. For example, high values for the Broken Count metric can indicate that files have been moved or that relative links in the application are broken.

4.6.3 [HTTP Step] Broken URL Content

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

4.6.4 [HTTP Step] DNS Time

This metric is not supported for this version of Enterprise Manager.

4.6.5 [HTTP Step] First Byte Time (ms)

Enterprise Manager breaks down each step element. Performance metrics for each step element can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.6.6 [HTTP Step] First Byte TIme per Page Element (ms)

This is the First Byte Time divided by the number of step elements.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

4.6.7 [HTTP Step] HTML Time (ms)

Enterprise Manager breaks down each step. Performance metrics for each step element can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

4.6.8 [HTTP Step] Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

4.6.9 [HTTP Step] Perceived Slowest Page Element Time (ms)

The amount of time that it would take a Web browser to play the slowest step element. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or a step that is slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.6.10 [HTTP Step] Perceived Time per Page Element (ms)

The average amount of time that it would take a Web browser to play each step element. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the step group to isolate and repair the problem.

4.6.11 [HTTP Step] Perceived Total Time (ms)

The amount of time that it would take a Web browser to play the step element. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem transactions. After you identify a step group that's slow to respond to user requests, you can drill down and analyze each phase of the step to isolate and repair the problem.

4.6.12 [HTTP Step] Redirect Time (ms)

Enterprise Manager breaks down each step into individual phases. Performance metrics for each phase of the step can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a step. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

4.6.13 [HTTP Step] Status

Indicates whether the Web transaction was successful.

4.6.14 [HTTP] Status Description

If the beacon is unable to run the transaction successfully, this metric returns a description of the error that prevented the transaction from running.

4.6.15 [HTTP Step] Time per Connection (ms)

This is the Connect Time divided by the number of connections made while playing a step.

User Action

Slow Time per Connection has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner. Transactions that use HTTPS will typically have a much higher Time per Connection than transactions that use HTTP.

4.6.16 [HTTP Step] Total Time (ms)

Indicates the overall time spent in processing the step. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

4.6.17 [HTTP Step] Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

4.6.18 [HTTP Step] URL

This is the URL associated with the step.

4.7 ICMP Echo Response Metrics

The following sections lists the ICMP Echo Response metrics, their descriptions, and user actions.

4.7.1 [ICMP Ping] Last Host

Indicates the last node that was reached successfully while traversing to the final destination.

User Action

If the last host is not your destination node, there may be an indication that network problems exist between the last host and the destination node. Validate that the host is up and that none of your routers are down.

4.7.2 [ICMP Ping] Number of Hops

Indicates the number of network nodes traversed to reach the host.

User Action

If this number is higher than you think it should be, examine your network configuration. Your routers may be routing packets improperly.

4.7.3 [ICMP Ping] Packets Dropped (%)

Indicates the percentage of packets that could not reach their destination.

User Action

Packets are usually dropped due to a congested network. Remove the source of the congestion or upgrade your network bandwidth.

4.7.4 [ICMP Ping] Response Time (ms)

Indicates the average amount of time that the agent waited before receiving a response for each "ping" sent to the host.

User Action

Slow response time could indicate that there is some network congestion or that a packet takes a long time to reach the host. Investigate your network configuration. When Response Time is high, the Number of Hops is usually also high.

4.7.5 [ICMP Ping] Status

Indicates that the host is reachable from the agent.

4.8 IMAP Response Metrics

The following table lists the IMAP Response metrics and their descriptions.

Table 4–2 IMAP Response Metrics

Metric	Description
[IMAP] Connect Time (ms)	Time it took (in milliseconds) to open an IMAP connection
[IMAP] Login Time (ms)	Time it took (in milliseconds) to log into an IMAP Service
[IMAP] Status	Current status of the IMAP service, either Up, Down, Status Pending, or Agent Unreachable
[IMAP] Time to Read Email (ms)	Time it took (in milliseconds) to read an e-mail message
[IMAP] Time to List Folders (ms)	Time it took (in milliseconds) to list the e-mail folders
[IMAP] Timing (ms)	Total time it took (in milliseconds) to open an IMAP connection, log into the IMAP service, list the e-mail folders, and read an e-mail message

4.9 LDAP Response Metric

The following section lists the LDAP Response metric and its description.

4.9.1 [LDAP] Status

Shows the current status of the LDAP service, either Up, Down, Status Pending, or Agent Unreachable.

4.10 NNTP Response Metrics

The following table lists the NNTP Response metrics and their descriptions.

Table 4–3 IMAP Response Metrics

Metric	Description
[NNTP] Connect Time (ms)	Time it took (in milliseconds) to open an NNTP connection
[NNTP] Status	Current status of the NNTP service, either Up, Down, Status Pending, or Agent Unreachable
[NNTP] Time to post news article (ms)	Time it took (in milliseconds) to post a message to the news group through the NNTP service
[NNTP] Total Time	Total time it took (in milliseconds) to open an NNTP connection, log in, retrieve a message and post a message to the news group
[NNTP] Time to retrieve news article (ms)	Time it took (in milliseconds) to retrieve a message from the NNTP service

4.11 OS Response Metrics

The following table lists the OS Response metrics and their descriptions.

Table 4–4 OS Response Metrics

Metric	Description
[Custom Script] Number of Retries	Total number of retries before the script is successfully executed
[Custom Script] Status	Status of the service test. The test is successful of the return code of the script is 0.
[Custom Script] Total Time (ms)	Total time required to run the script
[Custom Script] Custom Metric 1	Numeric value should be generated for each line of the custom script. This column corresponds to the first line generated by the script.
[Custom Script] Custom Metric 2	Numeric value should be generated for each line of the custom script. This column corresponds to the second line generated by the script.
[Custom Script] Custom Metric 3	Numeric value should be generated for each line of the custom script. This column corresponds to the third line generated by the script.
[Custom Script] Custom Metric 4	Numeric value should be generated for each line of the custom script. This column corresponds to the fourth line generated by the script.
[Custom Script] Custom Metric 5	Numeric value should be generated for each line of the custom script. This column corresponds to the fifth line generated by the script.
[Custom Script] Custom Metric 6	Numeric value should be generated for each line of the custom script. This column corresponds to the sixth line generated by the script.
[Custom Script] Custom Metric 7	Numeric value should be generated for each line of the custom script. This column corresponds to the seventh line generated by the script.
[Custom Script] Custom Metric 8	Numeric value should be generated for each line of the custom script. This column corresponds to the eighth line generated by the script.
[Custom Script] Custom Metric 9	Numeric value should be generated for each line of the custom script. This column corresponds to the ninth line generated by the script.
[Custom Script] Custom Metric 10	Numeric value should be generated for each line of the custom script. This column corresponds to the last line generated by the script.

4.12 POP Response Metrics

The following table lists the POP Response metrics and their descriptions.

Table 4–5 POP Response Metrics

Metric	Description
[POP] Connect Time (ms)	Time it took (in milliseconds) to open a POP connection
[POP] Login Time (ms)	Time it took (in milliseconds) to log into the POP service
[POP] Status	Current status of the POP service, either Up, Down, Status Pending, or Agent Unreachable
[POP] Time to Read Email (ms)	Time it took (in milliseconds) to read a short e-mail message
[POP] Timing (ms)	Total time it took (in milliseconds) to open a POP connection, log in, and read a short e-mail message

4.13 Port Checker Metrics

The following sections list the Port Checker metrics, their descriptions, and user actions.

4.13.1 [Port Checker] Status

Indicates whether agent could successfully connect to the Expected Open Ports or it could not connect to the Expected Closed Ports.

4.13.2 [Port Checker] Unexpectedly Closed Ports

Set of ports that were unexpectedly closed.

User Action

Check that there is no firewall blocking these ports. Check that the server listening on these ports is up.

4.13.3 [Port Checker] Unexpectedly Open Ports

Set of ports that were unexpectedly open.

User Action

If you have a firewall blocking these ports, check your firewall configuration.

4.14 SMTP Response Metrics

The following table lists the SMTP Response metrics and their descriptions.

Table 4–6 SMTP Response Metrics

Metric	Description
[SMTP] Connect Time (ms)	Time it took (in milliseconds) to open an SMTP connection
[SMTP] Status	Current status of the SMTP service, either Up, Down, Status Pending, or Agent Unreachable
[SMTP] Time To Send Email (ms)	Time it took (in milliseconds) to send a short e-mail message
[SMTP] Total Time (ms)	Total time it took (in milliseconds) to open an SMTP connection and send a short e-mail message

4.15 SOAP Response Metrics

The following sections list the SOAP Response metrics and their descriptions.

4.15.1 SOAP Response Time

Time taken by the beacon to complete the entire operation. This includes the time taken to send the HTTP request and receive the response.

4.15.2 SOAP Response Response Time (ms)

Time taken by the beacon to complete the entire operation. This includes the time taken to send the HTTP request and receive the response.

4.15.3 SOAP Response Status

This can be 0 (status down) or 1 (status up). The status is down when there is any error detected by beacon while performing the operation. The Status may be down in following cases:

- Due to any HTTP error
- No HTTP errors but due to a SOAP fault in the response

4.15.4 SOAP Status

This can be 0 (status down) or 1 (status up). The status is down when there is any error detected by beacon while performing the operation. The Status may be down in following cases:

- Due to any HTTP error
- No HTTP errors but due to a SOAP fault in the response

4.16 Oracle SQL Response

The following sections list the SQL Response metrics, their descriptions, and user actions.

4.16.1 [SQL] Close Time (ms)

Time taken to close the connection.

User Action

Close Time might be slow if the network performance is slow. Examine your network configuration.

4.16.2 [SQL] Connect Time (ms)

Total time taken to connect to the database.

User Action

Connect Time might be slow if the network is congested, if the database is having trouble authenticating the user, or if the database is having trouble allocating connections. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.16.3 [SQL] Execute Time (ms)

Time taken to execute the SQL statement.

User Action

Execute Time will be slow if the database performance is slow. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.16.4 [SQL] Fetch Time (ms)

Time taken to retrieve data from the server.

User Action

Fetch Time might be slow if the network bandwidth is low or if database performance is slow. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.16.5 [SQL] Fetch Time per Row (ms)

Time taken to fetch each row (Fetch Time / Number of Rows Fetched). Fetch Time per Row is a good metric to use for setting thresholds and for comparing the performance of different Oracle SQL Timing tests.

User Action

Use the Fetch Time per Row metric to identify data transfer problems. If the value of this metric is high, then the data-transfer bandwidth between the client and the database is poor. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.16.6 [SQL] Number of Rows Fetched

The total number of rows fetched during a query.

User Action

If the Number of Rows Fetched is not what you expected, then examine the contents of your database. Unexpected rows in the result could affect Fetch Time and Total Time.

4.16.7 [SQL] Prepare Time (ms)

Time taken to prepare the SQL statement. This usually includes fetching metadata for the object types in the query.

User Action

Prepare Time might be slow if the network performance is slow or if database performance is slow. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.16.8 [SQL] Status

Indicates whether the SQL or PL/SQL statement could be successfully executed.

4.16.9 [SQL] Status Description

Provides a description of the status.

4.16.10 [SQL] Total Time (ms)

The total time taken to connect to the database and run the query. It is the sum of Connect Time, Prepare Time, Execute Time, Fetch Time and Close Time.

User Action

Use the Total Time metric to identify database connectivity problems. Examine the other metrics to isolate and repair the problem. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.16.11 [SQL] Total Time per Row (ms)

Time taken to perform the entire test divided by the number of rows fetched (Total Time / Number of Rows Fetched). Total Time per Row is a good metric to use for setting thresholds and for comparing the performance of different Oracle SQL Timing tests.

User Action

Use the Total Time per Row metric to identify database connectivity problems. As with the Total Time metric, you should examine the other metrics to isolate and repair the problem. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database.

4.17 TNS Ping Response

The following sections list the TNS Ping Response metrics, their descriptions, and user actions.

4.17.1 [TNS] Average Response Time (ms)

Indicates whether the database responds to the pings.

4.17.2 [TNS] Pings Dropped (%)

Indicates the percentage of pings that did not receive a response.

User Action

Check that the network is not congested and that the database is not under heavy load. If you have an Enterprise Manager target instance for the database, you should consult the homepage for the database

4.17.3 [TNS] Status

Indicates whether the database responds to the pings.

5

Web Application Metrics

This chapter describes the Web Application metrics.

5.1 HTTP Content

The following sections lists the HTTP Content metrics, their descriptions, and user actions.

5.1.1 Average Connect Time

This metric measures the average connect time for all pages in the transaction. This is calculated as: Total Connect Time / Number of Connections Made. The Connect Time is one of the phases of a transaction that can help you isolate and fix response time problems.

User Action

The average connect time, when reviewed over a period of time, can indicate whether network congestion or other connectivity issues are the cause of poor Web application response time.

5.1.2 Average First Byte Time

This metric measures the average First Byte Time for all pages in the transaction. This metric is computed as: Total First Byte Time / Number of Requests Made (either to fetch HTML or content). The First Byte time is one of the phases of a transaction that can help you isolate and fix response time problems.

User Action

The average First Byte Time, when reviewed over a period of time, can indicate whether network congestion or other connectivity issues are the cause of poor Web application response time.

5.1.3 Average Response Time

A single transaction often accesses multiple Web pages. The Average Page Response metric calculates the average response time of the pages within a single transaction. This metric is calculated as: Total Transaction Time / Number of Pages in the Transaction. For example, if the transaction connects to four different Web pages, this metric will calculate the average response time for the four pages each time the transaction is run.

User Action

If a particular transaction continuously exceeds the Average Page Response threshold, use the Beacon Data page to test the transaction from other beacons and over a specific time period. Use this data to pinpoint any trends or specific beacons that generate the alerts. Display the Beacon Data page by clicking the value of a metric on the Transaction Performance Page.

5.1.4 Beacon Name

The beacon name if the name of the beacon for which the current metric data is being collected.

5.1.5 Broken Content

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.1.6 Broken Count

This metric measures the number of errors encountered when displaying content for the pages accessed by the transaction. For example, missing GIF images or style sheets will increase the value of the Broken Count metric.

User Action

Use this metric to measure the quality of the pages being served by your Web application. For example, high values for the Broken Count metric can indicate that files have been moved or that relative links in the application are broken.

5.1.7 Computed Response Time

This metric represents the estimated response time for a client such as a browser, to fetch all the pages in a transaction. The computed response time is calculated as if the contents of every page (such as images and HTML style sheets) were fetched in parallel using multiple threads.

User Action

Use the Calculated Response Time to predict the response time that will be experienced by your average end user.

5.1.8 Connect Time

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Connect Time is the first phase of a transaction and represents the time it takes for a connection to the Web server to be established for all requests.

User Action

Slow connect time has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner.

5.1.9 Content Time

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the Content Time, which is the amount of time taken to transfer page content to the browser. Page content includes images and style sheets, as opposed to the HTML coding for the page.

User Action

Consider reducing the size of images or other contents of the page

5.1.10 DNS Time

This metric is not supported for this version of Enterprise Manager

5.1.11 First Byte TIme

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser.

5.1.12 HTML Bytes

This metric provides information about the amount of data transferred during the selected transaction. For each transaction, this metric provides the total bytes of HTML code transferred from all the Web pages accessed by the transaction.

User Action

This metric can help you isolate the cause of any performance problems identified by this transaction. Be sure to consider the number of HTML bytes when you compare the response time of your Web Application transactions.

5.1.13 HTML Content

This metric serves as a container for a set of metrics that provide you with information about the content of the Web pages you are monitoring, as well as response time information.

5.1.14 HTTP Response

This metric is a container for a set of metrics you can use to measure the performance of your Web Application transactions. It indicates how quickly the pages respond to user requests.

5.1.15 HTML Time

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content, for example.

User Action

Slow HTML time could indicate problems with your HTML coding. Check the source code for your Web Application page to see if there are ways to streamline or improve the logic of the HTML code.

5.1.16 Page Content Bytes

This metric provides information about the amount of data transferred during the selected transaction. For each transaction, this metric provides the number of bytes that represent page content such as images and style sheets.

User Action

This metric can help you isolate the cause of any performance problems identified by this transaction. Be sure to consider the total number of bytes when you compare the response time of your Web Application transactions. Pages with many images or complex style sheets will return a high value for the Page Content Bytes metric.

5.1.17 Page Content Count

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.1.18 Redirect Count

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.1.19 Redirect Time

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a transaction. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request removes the possibility that the page content or the Web application software is causing a slow response time alert. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

5.1.20 Request Count

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.1.21 Slowest Response Time

A single transaction often accesses multiple Web pages. This metric indicates the maximum response time measured for a particular page within a transaction. The slowest page response time can be monitored for a specific transaction and from a specific beacon over a period of time.

You can set a threshold for this metric so that Enterprise Manager will generate an alert if the slowest page response for a particular transaction exceeds a value you specify when it is run from a specific beacon.

5.1.22 Status

This metric returns a value of 1 if the selected beacon was successfully able to run the transaction for this Web application target.

User Action

There are several possible causes to a failed transaction. First, check the availability of the Web application and host for the Web application target.

Second, check the availability of the Oracle Agent for this beacon.

5.1.23 Status Description

If the beacon is unable to run the transaction successfully, this metric returns a description of the error that prevented the transaction from running.

User Action

If you are reviewing the metric results from the All Metrics page, review the Value column of the Status Description table. The error description should offer clues about why the transaction failed.

5.1.24 Total Bytes

This metric provides information about the amount of data transferred during the selected transaction. For each transaction, this metric provides the total number of bytes transferred from all the Web pages accessed by the transaction.

User Action

This metric can help you isolate the cause of any performance problems identified by this transaction. Be sure to consider the total number of bytes when you compare the response time of your Web Application transactions.

5.1.25 Total Response TIme

Total transaction time indicates the overall time spent to process the transaction. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Content Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the total response time metric to identify problem pages. After you identify a page or transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.1.26 Transaction Name

The transaction name is the name of the transaction for which the current metric data is collected.

5.1.27 Transfer Rate

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

5.1.28 Web Application

You can use Oracle Enterprise Manager to view performance and availability metrics for your Web applications.

5.2 HTTP Step Group

The following sections lists the HTTP Step Group metrics, their descriptions, and user actions.

5.2.1 [HTTP Step Group] Broken URL Count

This metric measures the number of errors encountered when displaying content for the pages accessed by the step group. For example, missing GIF images or style sheets will increase the value of the Broken Count metric.

User Action

Use this metric to measure the quality of the pages being served by your Web application. For example, high values for the Broken Count metric can indicate that files have been moved or that relative links in the application are broken.

5.2.2 [HTTP Step Group] Broken URL Details

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.2.3 [HTTP Step Group] Connect Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the step group can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spend connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

5.2.4 [HTTP Step Group] DNS Time

This metric is not supported for this version of Enterprise Manager.

5.2.5 [HTTP Step Group] First Byte Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.2.6 [HTTP Step Group] First Byte TIme per Page (ms)

This is the First Byte Time divided by the number of pages in the step group.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.2.7 [HTTP Step Group] HTML Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

5.2.8 [HTTP Step Group] Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network

bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

5.2.9 [HTTP Step Group] Perceived Slowest Page Time (ms)

The amount of time that it would take a Web browser to play the slowest page in a step group. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that's slow to response respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem

5.2.10 [HTTP Step Group] Perceived Time per Page (ms)

The average amount of time that it would take a Web browser to play each page in the step group. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the step group to isolate and repair the problem.

5.2.11 [HTTP Step Group] Perceived Total Time (ms)

The amount of time that it would take a Web browser to play the step group. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem transactions. After you identify a step group that's slow to respond to user requests, you can drill down and analyze each phase of the step group to isolate and repair the problem.

5.2.12 [HTTP Step Group] Redirect Time (ms)

Enterprise Manager breaks down each step group into individual phases. Performance metrics for each phase of the step group can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a step group. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

5.2.13 [HTTP Step Group] Status

Indicates whether the Web transaction was successful.

5.2.14 [HTTP Step Group] Status Description

If the beacon is unable to run the transaction successfully, this metric returns a description of the error that prevented the transaction from running.

5.2.15 [HTTP Step Group] Time per Connection (ms)

This is the Connect Time divided by the number of connections made while playing a step group.

User Action

Slow Time per Connection has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner. Transactions that use HTTPS will typically have a much higher Time per Connection than transactions that use HTTP.

5.2.16 [HTTP Step Group] Total Time (ms)

Indicates the overall time spent in processing the step group. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.2.17 [HTTP Step Group] Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

5.3 HTTP Transaction

The following sections lists the HTTP Transaction metrics, their descriptions, and user actions.

5.3.1 [HTTP Transaction] Connect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spend connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many

connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

5.3.2 [HTTP Transaction] DNS Time

This metric is not supported for this version of Enterprise Manager.

5.3.3 [HTTP Transaction] First Byte Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.3.4 [HTTP Transaction] First Byte Time per Page (ms)

This is the First Byte Time divided by the number of pages in the transaction.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.3.5 [HTTP Transaction] HTML Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

5.3.6 [HTTP Transaction] Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

5.3.7 [HTTP Transaction] Perceived Slowest Page Time (ms)

The amount of time that it would take a web browser to play the slowest page in the transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that's slow to response respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.3.8 [HTTP Transaction] Perceived Time per Page (ms)

The average amount of time that it would take a Web browser to play each page in the transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.3.9 [HTTP Transaction] Perceived Total Time

The amount of time that it would take a web browser to play the transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.3.10 [HTTP Transaction] Redirect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a transaction. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

5.3.11 [HTTP Transaction] Status

Indicates whether the Web transaction was successful.

5.3.12 [HTTP Transaction] Status Description

If the beacon is unable to run the transaction successfully, this metric returns a description of the error that prevented the transaction from running.

5.3.13 [HTTP Transaction] Time per Connection (ms)

This is the Connect Time divided by the number of connections made while playing a transaction.

User Action

Slow Time per Connection has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner. Transactions that use HTTPS will typically have a much higher Time per Connection than transactions that use HTTP.

5.3.14 [HTTP Transaction] Total Time (ms)

Indicates the overall time spent to process the transaction. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a r transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.3.15 [HTTP Transaction] Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

5.4 HTTP User Action

The following sections lists the HTTP User Action metrics, their descriptions, and user actions.

5.4.1 [HTTP Step] Broken URL Content

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.4.2 [HTTP Step] Connect Time (ms)

Enterprise Manager breaks down each step element. Performance metrics for each step element can help you pinpoint the cause of a slow response time alert. Connect Time is

the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spend connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

5.4.3 [HTTP Step] DNS Time

This metric is not supported for this version of Enterprise Manager.

5.4.4 [HTTP Step] First Byte Time (ms)

Enterprise Manager breaks down each step into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. This metric measures the First Byte Time, which is the total time taken between the last byte of the request sent and the first byte of the response received by the server for all requests made. This includes the network latency and the time for the server to respond.

User Action

As with the Connect Time and Redirect Time, this metric can help you pinpoint whether or not the page content or Web application software is causing the slow response time, as opposed to the actual time it takes to transfer one byte of information to the browser. A high First Byte Time suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.4.5 [HTTP Step] First Byte Time per Page Element (ms)

This is the First Byte Time divided by the number of step elements.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.4.6 [HTTP Step] HTML Time (ms)

Enterprise Manager breaks down each step element. Performance metrics for each step element can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

5.4.7 [HTTP Step] Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

5.4.8 [HTTP Step] Perceived Slowest Page Element Time (ms)

The amount of time that it would take a Web browser to play the slowest step element. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that's slow to response respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem

5.4.9 [HTTP Step] Perceived Time per Page Element (ms)

The average amount of time that it would take a Web browser to play each page in a step. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the step group to isolate and repair the problem.

5.4.10 [HTTP Step] Perceived Total Time (ms)

The amount of time that it would take a Web browser to play the step element. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem transactions. After you identify a step group that's slow to respond to user requests, you can drill down and analyze each phase of the step to isolate and repair the problem.

5.4.11 [HTTP Step] Redirect Time (ms)

Enterprise Manager breaks down each step element. Performance metrics for each step element can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a step. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

5.4.12 [HTTP Step] Status

Indicates whether the Web transaction was successful.

5.4.13 [HTTP Step] Status Description

If the beacon is unable to run the transaction successfully, this metric returns a description of the error that prevented the transaction from running.

5.4.14 [HTTP Step] Time per Connection (ms)

This is the Connect Time divided by the number of connections made while playing a step element.

User Action

Slow Time per Connection has nothing to do with the content of the page itself. It is likely caused by a slow network or a busy Web server, which prevents the request from getting to the Web server in a timely manner. Transactions that use HTTPS will typically have a much higher Time per Connection than transactions that use HTTP.

5.4.15 [HTTP Step] Total Time (ms)

Indicates the overall time spent in processing the step. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.4.16 [HTTP Step] Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

Slow transfer rate can be caused by network congestion or other connectivity issues.

5.4.17 [HTTP Step] URL

This is the URL associated with the step.

5.5 HTTP Raw

The following sections lists the HTTP Raw metrics, their descriptions, and user actions.

5.5.1 HTTP Raw Broken URL Details

This metric is not currently collected by Oracle Enterprise Manager and is for internal use only.

5.5.2 HTTP Raw Connect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction, step or step group can help you pinpoint the cause of a slow response time alert. Connect Time is the total time spent in the transaction connecting to the server. There may be multiple connections made during a transaction. Time spent connecting for requests that result in redirects count as Redirect Time rather than Connect Time.

User Action

Significant Connect Time values are usually caused by a slow network or a busy Web server. Significant Connect Time values may also indicate that there are too many connections made during the transaction. Consider enabling HTTP persistent connections if the application does not already have them enabled.

5.5.3 HTTP Raw DNS Time

This metric is not supported for this version of Enterprise Manager.

5.5.4 HTTP Raw First Byte Time (ms)

This is the First Byte Time divided by the number of pages in the step, step group, or transaction.

User Action

A high First Byte Time per Page suggests that there may be high network latency between the agent and the service. Some applications generate an entire page before sending the first byte of that page. For such applications, a high First Byte Time could also indicate that the servers are taking a long time to generate each page.

5.5.5 HTTP Raw HTML Time (ms)

Enterprise Manager breaks down each step, step group, or transaction into individual phases. Performance metrics for each phase can help you pinpoint the cause of a slow response time alert. This metric measures the HTML Time, which is the amount of time it takes to transfer the HTML coding of the page to the browser. This metric does not include the time spent transferring images or other page content.

User Action

Slow HTML time could indicate that the application is taking a long time to finish generating each page. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low.

5.5.6 HTTP Raw Non-HTML Time (ms)

This is the amount of time it takes to transfer the non-HTML content such as images to the browser.

User Action

Slow Non-HTML time could indicate that the application is taking a long time to generate images. Alternatively, slow HTML time could indicate that network bandwidth between the agent and the service is low. Consider reducing the number of distinct images in the application.

5.5.7 HTTP Raw Perceived Slowest Page / Page Element Time (ms)

The amount of time that it would take a web browser to play the slowest page in the step, step group, or transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be.

User Action

Use this metric to identify problem pages. After you identify a page or transaction that's slow to response respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.5.8 HTTP Raw Perceived Time per Page / Page Element (ms)

The average amount of time that it would take a Web browser to play each page in the step, step group, or transaction. This is a good metric for setting thresholds because it is the closest active measurement of what the user-experience is likely to be. Because it is normalized on a per-page basis, Perceived Time per Page is also a good metric for comparing the relative performance of different transactions.

User Action

Use this metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.5.9 HTTP Raw Perceived Total Time (ms)

Indicates the overall time spent to process the step, step group, or transaction. This includes all the phases of the step / step group / transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.5.10 HTTP Raw Redirect Time (ms)

Enterprise Manager breaks down each transaction into individual phases. Performance metrics for each phase of the transaction can help you pinpoint the cause of a slow response time alert. Some pages automatically redirect the HTTP request to another page. Redirect time represents the total time of all redirects within a transaction. The time taken to redirect the request can affect the overall response time of the page.

User Action

Significant time taken to redirect the HTTP request. If the redirect is causing the performance problems, consider alternative solutions to sending the user to another HTML page.

5.5.11 HTTP Raw Status

Indicates whether the Web transaction was successful.

5.5.12 HTTP Raw Status Description

If the beacon is unable to run the step, step group, or transaction successfully, this metric returns a description of the error that prevented the transaction from running.

5.5.13 HTTP Raw Time Per Connection

This metric measures the average connect time for all pages in the transaction. This is calculated as: Total Connect Time / Number of Connections Made. The Connect Time is one of the phases of a transaction that can help you isolate and fix response time problems.

User Action

The average connect time, when reviewed over a period of time, can indicate whether network congestion or other connectivity issues are the cause of poor Web application response time.

5.5.14 HTTP Raw Transfer Rate (KB per second)

The transfer rate indicates how quickly data is being transferred from the Web server to the client browser. This is computed as: Total Kilobytes Received / Total Transaction Time.

User Action

Slow transfer rate can be caused by network congestion or other connectivity issues.

5.5.15 HTTP Raw Total Time (ms)

Indicates the overall time spent to process the step, step group, or transaction. This includes all the phases of the transaction, including Connect Time, Redirect Time, First Byte Time, HTML Time, and Non-HTML Time. This metric calculates total transaction time by assuming all contents of a page are fetched in a serial manner.

User Action

Use the Total Time Metric to identify problem transactions. After you identify a r transaction that's slow to respond to user requests, you can drill down and analyze each phase of the transaction to isolate and repair the problem.

5.5.16 HTTP Raw URL

This is the URL associated with the step.