



**Hewlett Packard**  
Enterprise

## **NonStop Database Analyzer Repository Guide**

### **Abstract**

This guide provides information about the NonStop Database Analyzer Repository.

## Notices

The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Links to third-party websites take you outside the Hewlett Packard Enterprise website. Hewlett Packard Enterprise has no control over and is not responsible for information outside the Hewlett Packard Enterprise website.

## Acknowledgments

Intel®, Itanium®, Pentium®, Xeon®, Intel Inside®, and the Intel Inside logo are trademarks of Intel Corporation in the U.S. and other countries.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

Java® and Oracle® are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.



# Contents

- New and changed information..... 6**
  - New and changed information in the P13181-003 edition..... 6
  - New and changed information in the P13184-002a edition..... 6
  - New and changed information in the P13184-002 edition..... 6
  
- NSDA repository..... 7**
  - Accessing the repository..... 7
  - Controlling access to the repository..... 7
  
- NSDA repository design..... 8**
  - Data model..... 8
    - Configuration Tables..... 8
    - Dimension Tables..... 9
    - Fact and Summary tables..... 9
    - Views..... 10
  - Naming conventions..... 10
    - Naming convention for tables..... 11
    - Naming convention for views..... 11
  - Header columns..... 13
    - All Fact tables..... 13
    - Metric Fact tables..... 15
    - Event Fact tables..... 15
    - Health Fact Views..... 15
  
- Repository maintenance..... 19**
  
- Repository tables..... 20**
  - Common headers..... 20
  - Table definitions..... 20
  
- Configuration tables..... 21**
  - configuration\_instance\_table..... 21
  - configuration\_maintenance\_table..... 23
  - configuration\_node\_table..... 23
  - configuration\_report\_definitions\_table..... 24
  
- Configuration views..... 26**
  - configuration\_node\_latest..... 26
  - configuration\_service\_dataservice..... 26
  
- Dimension tables..... 28**



dimension_component_table.....	28
dimension_health_guidance_table.....	28
dimension_healthtype_table.....	28
dimension_instance_bitmap_table.....	28
dimension_instance_role_table.....	29
dimension_instance_table.....	29
dimension_object_subtype_table.....	29
dimension_object_type_table.....	29
dimension_publication_type_table.....	30
dimension_score_table.....	30
dimension_severity_table.....	30
dimension_statetype_table.....	31
dimension_subhealthtype_table.....	31
dimension_substatetype_table.....	31
dimension_subtype_type_table.....	32
dimension_user_association_table.....	32

## **Event tables.....33**

event_dsa_table.....	33
event_text_table.....	34

## **Health tables.....35**

health_change_table.....	35
health_overview_table.....	35
health_summary_table.....	35

## **Metric tables.....37**

metric_diskio_summary_table.....	37
metric_diskio_table.....	37
metric_heat_summary_table.....	38
metric_jobquery_summary_table.....	38
metric_jobquery_table.....	39
metric_jobquery_topncputime_table.....	40
metric_jobquery_topndiskios_table.....	41
metric_jobquery_topnmemory_table.....	42
metric_jobquery_topnreads_table.....	43
metric_topnrows_table.....	43
metric_jobquery_topnrows_table.....	44
metric_jobquery_topnruntime_table.....	45
metric_jobquery_topnwrites_table.....	45
metric_loadqueue_summary_table.....	46
metric_loadqueue_table.....	47
metric_memory_summary_table.....	47
metric_memory_table.....	48
metric_network_summary_table.....	50
metric_network_table.....	51
metric_nonstop_query_plan_table.....	52
metric_nonstop_query_progress_table.....	53
metric_nonstop_query_text_table.....	60
metric_nonstop_session_table.....	61
metric_nonstop_tenant_access_table.....	61
metric_nonstop_tenant_table.....	62



metric_nonstop_user_table.....	62
metric_physdiskstats_table.....	62

**Support and other resources..... 66**

Accessing Hewlett Packard Enterprise Support.....	66
Accessing updates.....	66
Remote support.....	67
Warranty information.....	67
Regulatory information.....	67
Documentation feedback.....	68



# New and changed information

## **New and changed information in the P13181-003 edition**

Updated content pertaining to NSDA 1.2.

## **New and changed information in the P13184-002a edition**

Updated the supported RVUs on the title page.

## **New and changed information in the P13184-002 edition**

Updated content pertaining to NSDA 1.1.



# NSDA repository

The NSDA Repository is a data warehouse for manageability data collected about data service instances (DSIs). Each central management instance (CMI) uses a NSDA SQL database that hosts the Repository. The data contained in the Repository enables point-in-time and comparative analysis, correlating information from multiple subject areas on a temporal basis.

## Accessing the repository

To access the repository, query the repository using the NSDA Query Whiteboard.

The repository views are provided in the following formats:

- **Table Views:** Best performing method. This view uses joins and combines configuration and dimension tables to provide human-friendly names for enumerations and IDs.
- **Versioned Views:** Guarantee downward compatibility when an NSDA table is changed – no such guarantee is provided for the underlying Repository tables. Version Views have a numeric suffix that represents the view version. Use Versioned Views to avoid changes to your reports due to an NSDA upgrade.
- **Base Views:** Provide the latest version of the view directly reflecting the underlying table. Use this class of views for on-demand queries and when exploring the repository data. Base Views have a "current" suffix.

Use the repository tables whenever possible to ensure best performance. Using views can significantly affect performance and must be avoided for queries that access few columns only and/or are used frequently.

The data in the repository is interleaved. Each view might return for several instances. Therefore, use the `instance_id` column in query predicates to ensure that your query returns information on a per-instance level.

## Controlling access to the repository

By default, all users have read access to the Repository views and tables. The Repository Database Administrator can adjust these permissions as required to fit site policy. However, make sure that the `<nsda-admin-id>` user has read and write access to the tables. Secure Repository views and tables so that all intended users can access them through NSDA queries. The logon ID given out for the Repository must have read privileges on the Fact and Summary views.



# NSDA repository design

In the NSDA Repository:

- Subject-oriented fact tables store manageability data in a time-sequenced manner.
- Dimensions are used to describe and join data stored in different fact tables.
- Dimensional and fact data are defined per a common set of conformational rules.
- Summary tables provide access to cross-instance subjects (for example, jobs/queries) in a simplified, conforming manner.

The data model is centered on a star schema combined with a specific intent to avoid snowflake schemas as much as possible to ensure a simplified query model.

Given the operational nature of manageability data, the primary goal of the Repository is to provide the means to perform ongoing analysis and reporting similar to that of an Operational Data Store (ODS).

To retrieve data from the Repository, query the tables and/or views described in this guide.

Performance data, health, and state information are available for, but not necessarily limited to:

- Access and connectivity services (for example: health, and performance data for ODBC services)
- Data Service (job/query performance data and health data)
- Infrastructure services (performance, and health and state data for name nodes, manageability software)
- Operating system (health data for various system services)
- Server (cpu and memory performance data)
- Storage (physical disks, MSAs, SSD health, and performance data)

The services running in the DSIs determine the types and timeframes for data visible in the repository. For example, if there are no NonStop SQL/MX instances using the repository, then no NonStop SQL/MX data is present.

Any user with appropriate privileges can issue queries against the Repository using the NSDA SQL Whiteboard function. Administrators with the appropriate grant privileges can, in turn, grant other users access to the Repository.

Security policies at your company determine the level of privileges required to access Repository tables and views. Obtain the privileges you need from your NSDA administrator.

## Data model

The Repository consists of a set of views and tables. They contain everything from facts to analyzed information and configuration information.

The DDL files for the repository are on the CMI in the `/opt/seapilot/etc/mi/repository` directory.

## Configuration Tables

Configuration Tables are used to define customization of the Repository. For example, configuration settings related to maintenance operations. In addition, Configuration Tables are used to define instancewide settings.

The Configuration Tables are designed to be used for a multi-instance repository. Therefore, each Configuration Table contains an instance ID, which enables the data to be joined with other tables for multi-instance support.



## Dimension Tables

Dimension Tables are used to provide descriptive textual representations of information stored in fact and summary tables. Such descriptions spans from simple textual translations of enumerated values to rich alternative descriptions that can be found in, for example, date dimensions.

The Dimension Tables are designed to be used for a multi-instance repository. Therefore, each Fact and Summary Table contains an instance ID, which enables the data to be joined with the other tables for multi-instance support.

## Fact and Summary tables

Tables contain dimension, fact, summary (aggregation), and other data. The tables are updated continuously by the NSDA Application software.

### Fact tables

The following fact tables are used.

Table name	Description
Summary	High-level information about a subject area is normalized to present information about a given subject. The information is presented in a manner that is neutral to the different data services management by NSDA. On a case-by-case basis, Summary Fact Tables can be used to provide quick access to often-used information about a specific data service.
Detail	In-depth information about a subject area, either generic or data-service specific. A given subject area can be represented as single or multiple detail tables.
Current	A short-term periodic snapshot table used to optimize queries for live monitoring of a subject area. Generally, the Current Fact table is defined in the same manner as the corresponding Summary Table. The only difference between the Current Fact table and the Summary Fact table is that the Current Fact table contains information spanning minutes/hours while the Summary Fact table contains information spanning days/weeks.
Historic	Historical Fact tables contain long-term information for different subject areas. The data in the Historical Fact tables are copied from the Summary and Detail tables and can be aggregated for long-term use. Most Fact Tables are considered Historic Fact tables.

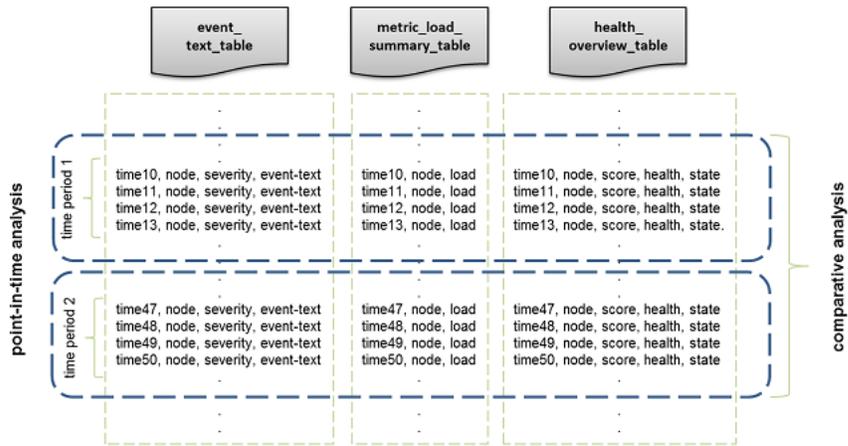
All fact tables share a common set of columns, which are referred to as "Header Columns." These columns serve the following purposes:

- A common set of identification columns enables both tenant interleaving and external multi-instance repositories plus identifier-to-text translation.
- A common set of date and time columns enable temporal analysis of all data.
- Where applicable, references to detail tables.

The remainder of each table contains Data Columns, which are defined per a common set of rules to ensure fact conformance.

The Fact Tables can be used to drill down into a time period and/or to compare information among several time periods. The following figure illustrates this concept:





The figure can be interpreted as follows:

- Fact tables contain information about different subject areas such as event, metric, and health.
- Each fact table records information using a conforming time stamp. A high resolution is used. So, it is not expected that the different tables contain exactly matching time stamps for the different rows.
- Point-in-time analysis obtains information for a given time range from the different tables. This gains information necessary to gain an understanding of the managed data service for that time range:
  - Were there any important events?
  - What was the instance load like?
  - Were there any health issues?

Depending on the answers to these questions, further information can be found in the detailed fact tables.

Comparative analysis can be performed by aggregating information (most often, metric data, but other aggregations such as summary error information/counts are also possible) for the different time periods. Once aggregated, the different time periods can be compared to, for example, understand the impact of a configuration change or to perform predictive growth analysis.

### Summary tables

Summary Fact Tables are used to provide normalization of underlying detailed data.

## Views

Views come in the following formats:

- **Base Views:** Provide the latest version of the view directly reflecting the underlying table. Use this class of views for on-demand queries and when exploring the Repository data.
- **Versioned Views:** Guarantee downward compatibility when an NSDA table is changed – no such guarantee is provided for the underlying Repository tables or base views. Therefore, clients wanting to query the Repository information must use the views and never access the tables directly.

## Naming conventions

Each schema follows a shared naming convention for tables and views. In addition, each table is defined per specific rules. The rules vary depending on whether the table is a dimensional table, a fact table, or a summary table.



## Naming convention for tables

All tables use the following naming convention, which is defined in lowercase except where noted:

```
{subject}[_data-service]{_qualifier}[_sub_qualifier]_table
```

Name part	Usage	Examples
subject	Describes the overall subject of the table. Subject is expressed in singular form. The subject part of the table name is mandatory.	<ul style="list-style-type: none"><li>• dimension</li><li>• event</li><li>• metric</li><li>• space</li></ul>
_data_service	For fact tables, identifies the table as containing data for a specific data service. The _data-service part of the table name is optional.	nonstop
_qualifier	For fact and summary tables, the _qualifier identifies the component name or an abbreviation. Abbreviations are expressed in uppercase. For dimensional tables, the _qualifier describes the purpose of the dimension. The _qualifier part of the table name is mandatory and is provided as a compound word.	Fact tables: <ul style="list-style-type: none"><li>• _node</li><li>• _service</li></ul> Dimension tables: <ul style="list-style-type: none"><li>• _component</li><li>• _date</li></ul>
_sub_qualifier	For fact and summary tables, the _sub-qualifier describes the purpose of the table. _sub-qualifier is normally not used for dimension tables, but follows the same rules as Fact and Summary tables. The _sub-qualifier part of the table name is optional. When used, the _sub-qualifier is specified as a compound word.	<ul style="list-style-type: none"><li>• _current</li><li>• _detail</li><li>• _summary</li></ul>
_table	Identifies that the name represents a table (as opposed to a view.) The _table part of the table name is mandatory.	_table

Examples:

- event\_text\_table
- metric\_jobquery\_summary\_table

## Naming convention for views

All views use the following naming convention, which is defined in lowercase except where noted:

```
{subject}[_data-service]{_qualifier}[_sub_qualifier][_version | _current]
```



<b>Name part</b>	<b>Usage</b>	<b>Examples</b>
subject	Matches the underlying fact/dimension table.	<ul style="list-style-type: none"> <li>• dimension</li> <li>• event</li> <li>• metric</li> <li>• space</li> </ul>
_data_service	Matches the underlying fact/dimension table.	nonstop
_qualifier	Matches the underlying fact/dimension table.	Fact tables: <ul style="list-style-type: none"> <li>• _node</li> <li>• _service</li> </ul> Dimension tables: <ul style="list-style-type: none"> <li>• _component</li> <li>• _date</li> </ul>
_sub_qualifier	Matches the underlying fact/dimension table.	<ul style="list-style-type: none"> <li>• _current</li> <li>• _detail</li> <li>• _summary</li> </ul>
_version	<p>A number identifying the version of the view.</p> <p>The first version uses 1, the second version uses 2, and so on. A new version of a view is created when the underlying table is updated. At the same time, current and previous versions of the view are updated to provide the same information as the views provided with the previous version of the underlying table.</p> <p>If present, then the view is a Version View.</p>	<ul style="list-style-type: none"> <li>• _1</li> <li>• _2</li> </ul>
_current	The current or Base view.	_current

Examples:

- event\_text\_1
- metric\_jobquery\_summary\_2



# Header columns

All fact and summary tables share a common set of columns, which are referred to as “Header Columns.” These columns serve the following purposes:

- A common set of conforming foreign keys provides a simple way to join data among views.
- A common set of identification columns enables both tenant interleaving and external multi-instance repositories.
- A common set of date and time columns enables temporal analysis of all data.

The remainder of each table contains Data Columns, which are defined per a common set of rules to ensure fact conformance.

Columns that are used as part of the primary key in a table are defined as “NOT NULL”. All other columns (header or data) are defined as “NULL.” This technique allows NSDA to write partial data to the Repository tables without affecting calculations and so on. In addition, the `instance_id` column is defined as “NOT NULL” because this column is required for joins.

## All Fact tables

The following defines the Header Columns that all Fact and Summary tables contain.

Column name	Data type	Default	Description
<code>instance_id</code>	integer unsigned	not null	A unique instance identifier. Identifies the instance that the data pertains to.  The instance is a logical representation of a managed entity; for example, a Hadoop cluster or a NSDA SQL database. Join this value to the <code>instance_id</code> column in the <code>configuration_instance_table</code> to provide a human-friendly name for the instance.
<code>tenant_id</code>	largeint	null	Catalog UID of the tenant database.  Tenant ID 0 indicates non-tenant database
<code>calendar_utc_date</code>	date	null	The date the data was generated.  In Coordinated Universal Time (UTC) format: <a href="http://en.wikipedia.org/wiki/Coordinated_Universal_Time">http://en.wikipedia.org/wiki/Coordinated_Universal_Time</a>

*Table Continued*



Column name	Data type	Default	Description
gen_utc_ts	timestamp(6)	not null	The timestamp when the data was generated in Coordinated Universal Time (UTC) format.
upsert_utc_ts	timestamp(6)	null	The time the data was last updated into the repository table in Coordinated Universal Time (UTC) format.
component_id	integer unsigned	null	A unique identifier identifying the component for which data is provided. Join this value with the component_id column in the dimension_component_table to provide a human-friendly name for the component.
process_id	Int	null	The process ID (from the operating system) for the data sensor that provided the data on the DSI. Mostly used for debugging purposes.
thread_id	integer unsigned	null	The thread ID for the data sensor that provided the data on the DSI. Mostly used for debugging purposes.
host_id	integer unsigned	null	Linux host ID; a numeric identifier for the node where the data originated.
ip_address	varchar(32)	null	The IP address of the node where the data originated.
service_name	varchar(32)	null	Logical name of service generating the data.
sequence_number	integer unsigned	null	Sequence number generated by the service that provided the data.  Used to detect whether a message is missing for a specific service.

Columns COMPONENT\_ID through PROCESS\_NAME indicate the source of the information for a row. Such information is important when analyzing performance issues, distribution, clustering, and so on, affecting the performance and/or function of NSDA.



## Metric Fact tables

For Metric Fact tables, the following are often added to the header columns.

Column name	Data type	Default	Description
requested_sampling_interval_ms	Int	null	For periodic publications only, requested sampling time interval (ms); otherwise null, such as metrics on-demand.
actual_sampling_interval_ms	Int	null	For periodic publications only, actual sampling interval achieved during execution (ms)

## Event Fact tables

For Event Fact tables (contains log messages and preindexed log messages), the following columns are added to the header columns.

Column name	Data type	Default	Description
event_id	Int	not null	Event identifier.
severity_enum	Int	null	Enumeration for the event severity. Join this value to the severity_id column in the dimension_severity_table to provide a human-friendly name for severity.
role_longname	varchar(128)	null	SQL role name. Null for data services that do not support SQL roles.

## Health Fact Views

For Health Fact tables, the following columns are added to the header columns.



Column name	Data type	Default	Description
publication_type	Int	not null	<p>Defines what type of health/state publication has been generated.</p> <p>0 - Baseline. A baseline publication is generated on a periodic interval to create a common baseline for all health/state data in the instance.</p> <p>1 - Incremental. An incremental publication is generated when a health and/or state change is detected for an object being checked.</p> <p>2 - Manual. A manual publication is generated when an operator runs the workload from a shell prompt passing the publication argument to the workflow.</p>
check_interval_secs	Int	null	<p>Describes the interval used for the verification in seconds. Combined with the gen_utc_ts fields, it is possible to use this value to estimate the next time the check workflow will be run.</p> <p>A value of NULL or 0 means that the check workflow was run manually and, therefore, it is impossible to calculate the next time the check workflow will run.</p>
error_enum	Int	null	<p>Indicates whether the workflow encountered an error if not NULL or 0.</p> <p>The value of this field represents an enumeration.</p>
error_text	varchar(128)	null	A short, descriptive textual representation of <error>.

For Health Fact Detail tables, the following columns are added to the header columns.

Column name	Data type	Default	Description
logical_object_type_enum	Int	not null	An enumeration of the object type as represented in the NSDA software. Join this value to the type_id column in the dimension_objecttype_table to provide a human-friendly name for the enumeration.
logical_object_subtype_enum	Int	null	An enumeration of the object subtype as represented in the NSDA software.
logical_object_subtype_name	varchar(32)	null	A human-friendly name for logical_object_subtype_enum. Join this value to the subtype_id column in the dimension_objectsubtype_table to provide a human-friendly name for the enumeration.

*Table Continued*



Column name	Data type	Default	Description
logical_object_name	varchar(128)	null	The name of the logical object.
logical_object_qual_1_longname	varchar(128)	null	A qualifier for the logical object name; first part.
logical_object_qual_2_longname	varchar(128)	null	A qualifier for the logical object name; second part.
logical_object_path_shortname	varchar(8)	null	Path qualifier of logical name of the object name.
physical_object_type_enum	Int	null	An NSDA enumeration of the object type as represented at the Linux level. Join this value to the type_id column in the dimension_objecttype_table to provide a human-friendly name for the enumeration.
physical_object_subtype_enum	Int	null	An NSDA enumeration of the subobject type as represented at the Linux level. Join this value to the subtype_id column in the dimension_objectsubtype_table to provide a human-friendly name for the enumeration.
physical_object_longname	varchar(128)	null	The name of the physical object.
current_state_enum	Int	null	NSDA normalized enumeration of current state of the object. Join this value to the state_enum column in the dimension_objectsubtype_table to provide a human-friendly name for the enumeration.
current_state_raw_name	varchar(32)	null	Current state before NSDA normalization; what the underlying component reports.
previous_state_enum	Int	null	NSDA normalized enumeration of previous state of the object. Join this value to the state_enum column in the dimension_objectsubtype_table to provide a human-friendly name for the enumeration.
previous_state_raw_name	varchar(32)	null	Previous state before NSDA normalization; what the underlying component reports.
state_change_utc_ts	timestamp(6)	null	The time a state change was detected in UTC (Universal Standard Time) format.

*Table Continued*



Column name	Data type	Default	Description
current_health_enum	Int	null	NSDA normalized enumeration of current health of the object. Join this value to the health_enum column in the dimension_healthtype_table to provide a human-friendly name for the enumeration.
current_health_raw_name	varchar(32)	null	Current health before NSDA normalization; what the underlying component reports.
previous_health_enum	Int	null	NSDA normalized enumeration of previous health of the object. Join this value to the health_enum column in the dimension_healthtype_table to provide a human-friendly name for the enumeration.
previous_state_raw_name	varchar(32)	null	Previous health before NSDA normalization; what the underlying component reports.
health_change_utc_ts	timestamp(6)	null	The time a health change was detected in UTC (Universal Standard Time) format.



# Repository maintenance

See the *NSDA Administrator Guide* for information about maintenance of the Repository.



# Repository tables

## Common headers

Each table ends with a set of common header columns. The columns allow data for different subject areas to be linked for a specific time period and grouped by predicates such as the DSI, and user that the data pertains to.

The header columns are not shown in the table definitions in the guide.

## Table definitions

The Metadata Queries category in the NSDA SQL Whiteboard provides reports that allow you to list all the tables (and views) in the Repository and to list each column in the table.

For table definitions, see the following subsections:

- **Configuration tables**
- **Dimension tables**
- **Event tables**
- **Health tables**
- **Metric tables**



# Configuration tables

The Configuration tables provide access to NSDA configuration settings.

## configuration\_instance\_table

This table defines the instances to the NSDA web application. It is populated as part of adding an instance to the NSDA application.

Column name	Data type	Description
instance_id	integer unsigned	A unique instance identifier. Identifies the instance that the data pertains to.  The instance is a logical representation of a managed entity. For example, a Hadoop cluster or a NSDA SQL database.
instance_longname	varchar(128)	A description of the instance used in displays and reports that provides more detail about the instance.
instance_role_enum	int	Identifies what type of instance the instance_id represents:  8 — NonStop SQL/MXData Service Instance
instance_type_name	varchar(32)	The type of data service being managed.  Must be specified in lowercase:  nonstop
instance_description_longtext	varchar(800)	A description of the instance used in displays and reports that provides more detail about the instance.
instance_contact_id	int	Reserved for future use.
time_zone_name	varchar(32)	The time zone the instance is running in. Currently set to ETC or UTC.

*Table Continued*



Column name	Data type	Description
time_format_enum	int	Indicates the format in which to display time: <ul style="list-style-type: none"> <li>• 1 = 12 hour format</li> <li>• 2 = 24 hour format</li> </ul>
instance_local_flag	int	Indicates that the repository is local to the monitoring instance. The repository that contains manageability data for this instance is embedded in the instance.
instance_version	varchar(64)	The version of the data service that is installed on this instance. For NonStop SQL/MX, the version might be T8010 L36 AOH or T8010 L37 (AOJ)+AOK.
nsda_user_name	varchar(32)	The Linux user name that NSDA runs as. It is the admin user ID.
nsda_guest_user_name	varchar(32)	The database identifier name of the nonadministrative ID used to verify user-level access to data service components.
restgenie_port_number	int	The RESTGenie port allows the NSDA Web Application to send commands to the instance. Two consecutive ports are reserved. The value specified is the lower value. This value is the first port (http port). The second port is the https port, which is this value + 1.
server_url_address	varchar(512)	The JDBC URL for the configured CMI or DSI. Used to connect to the instance from the Query Whiteboard.
primary_host_name	varchar(256)	The primary node for accessing this instance, where applicable. Not used at this point, so, NULL.
instance_display_mode	varchar(32)	Indicates if the instance must be displayed in the NSDA Web Application. If the flag is set to DISPLAY, it is included in drop-down/ selections.



## configuration\_maintenance\_table

This table defines aging policies for data in each repository table.

Used by `/opt/seapilot/etc/mi/repository/scripts/maintenance/rep_age_all.sh` when aging out data from the Repository. See the *NSDA Administrator Guide* for more details about data aging.

Column name	Data type	Description
table_id	int	Unique identifier for the repository table.
table_version_id	int	Version of the table.
table_longname	varchar(128)	Name of the table.
max_retention_days	int	Delete data older than this number of days.
retention_modifier	varchar(256)	A SQL query predicate to be applied to data aging beyond basic aging by date and by instance_id. For example, if data for jobs for a particular user_longname are to be maintained beyond the usual aging window, rather than aging automatically, the query predicate might include “where user_longname NOT like ‘finance_critical’;”

## configuration\_node\_table

Contains the configuration of the nodes that are associated with an instance. Configurations have particular times, so that all the nodes with the same `gen_utc_ts` are from the same configuration. Nodes might come and go from in the instance over time, meaning that the configuration for a particular time is best defined by selecting the set of nodes whose `gen_utc_ts` is maximally less than the time stamp at which the query is looking.

For a description of the columns in this table, see **Header columns**. The following additional columns exist beyond the Header Columns.

Column name	Data type	Description
node_longname	varchar(128)	Fully qualified domain
node_ip_address	varchar(32)	IP address of the host.
alternate_ip_address	varchar(32)	Alternate IP address of the host.

*Table Continued*



Column name	Data type	Description
node_hostid	varchar(32)	Host ID.
node_groupid	int	Host group ID.

## configuration\_report\_definitions\_table

This table stores the Query Whiteboard reports.

Column name	Data type	Description
unique_name	varchar(64)	Unique identifier for the report.
version_id	int	Version number of the report.
category_name	varchar(64)	Category name in the open dialog in the query whiteboard.
subcategory_name	varchar(64)	Subcategory name in the open dialog in the query whiteboard.
description_text	varchar(2048)	Help text for the report.
query_text	varchar(29000)	The report query and graphing instructions.
title_name	varchar(64)	The title of the report.
author_name	varchar(128)	NSDA User that created/updated the report.
clonedFrom_name	varchar(64)	Contains the unique name of the report from which this report was cloned. Currently not being used or set.
writeProtected_bool	boolean	Indicates if the report can be updated. Currently not used.
creation_lct_ts	timestamp(6)	Timestamp when this version of the report was created.
versionComment_text	varchar(255)	Comment describing the specific version of the report.

*Table Continued*



<b>Column name</b>	<b>Data type</b>	<b>Description</b>
query_type	char(1)	'S' indicates a system report. 'U' indicates a user-defined report.
isDeleted_bool	boolean	Only reports that have a 'false' value are fetched and displayed in the Whiteboard. So you can mark some reports as obsolete by setting the delete flag to true.
supported_instances_bitmap	int	An integer value that is a bitwise AND of the instance bitmap values of the instance types supported by the report.



# Configuration views

## configuration\_node\_latest

Provides the latest entry for each host within the last 30 minutes.

Column name	Data type	Description
instance_id	integer unsigned	The ID of the instance.
host_name	varchar(128)	The fully qualified domain name of the host.
ip_address	varchar(32)	The IP address of the host.
gen_utc_ts	timestamp(6)	The timestamp when the information was last updated.

## configuration\_service\_dataservice

Associates host to roles (control or data), service, and component for all data services.

Column name	Data type	Description
gen_utc_ts	timestamp(6)	The time stamp when the data was generated in Coordinated Universal Time (UTC) format.
calendar_utc_date	date	The calendar date the data was generated in Coordinated Universal Time (UTC) format.
instance_id	integer unsigned	A unique instance identifier. Identifies the instance that the data pertains to.
instance_longname	varchar(128)	A human-friendly name of the instance ID.
host_name	varchar(256)	Fully qualified domain name for the host.
ip_address	varchar(32)	The IP address for the host.
role_name	varchar(7)	Whether the node is a control or data node.

*Table Continued*



Column name	Data type	Description
service_name	varchar(9)	Name of the service, such as: <ul style="list-style-type: none"> <li>• NonStop</li> <li>• HDFS</li> <li>• MapReduce</li> <li>• NSDA SQL</li> <li>• Yarn</li> <li>• HBase</li> </ul>
component_name	varchar(32)	Name of the component, such as: <ul style="list-style-type: none"> <li>• NameNode</li> <li>• DataNode</li> <li>• HMaster</li> <li>• RegionServer</li> <li>• NSDA SQL-Database</li> </ul>



# Dimension tables

The Dimension tables provide mapping of enumerations to text and association information into groups.

## dimension\_component\_table

Maps component enumerations to component and service name. Component enumerations are used in many fact tables.

Column name	Data type	Description
component_id	int	Component enumeration.
component_name	varchar(32)	Textual name of component enumeration.
component_description_text	varchar(80)	Long description of component enumeration.
supported_instance_types	int	Bitmap of supported instance types.
service_name	varchar(32)	Name of service to which the component belongs.

## dimension\_health\_guidance\_table

This table is not used in the current version of NSDA.

## dimension\_healthtype\_table

Maps health enumerations to text. Health enumerations are used in health-oriented tables.

Column name	Data type	Description
health_enum	int	Health enumeration.
health_name	varchar(32)	Textual name of health enumeration.
health_description_text	varchar(80)	Long description of health enumeration.

## dimension\_instance\_bitmap\_table

Maps a bitmap to an instance type. Used to determine what reports and displays apply to a specific instance. Instance bitmaps are used in configuration\_report\_definitions\_table.



Column name	Data type	Description
instance_type_name	varchar(32)	Text representation of instance_type_bitmap.
instance_type_bitmap	int	Bitmap value: 64 — NonStop SQL/MX

## dimension\_instance\_role\_table

Maps instance-role enumerations to text. The instance role is used in configuration\_instance\_table.

Column name	Data type	Description
instance_role_enum	int	Enumeration of instance role.
instance_role_name	varchar(32)	Text representation of instance_role_enum.
instance_role_description_text	varchar(80)	Descriptive text for instance_role_enum.

## dimension\_instance\_table

This table is not used in the current version of NSDA.

## dimension\_object\_subtype\_table

Maps object subtype enumerations to text. Object subtypes are used in many fact tables.

Column name	Data type	Description
subtype_id	int	Enumeration of object subtype.
subtype_name	varchar(32)	Text representation of subtype_id.
subtype_description_text	varchar(80)	Descriptive text for subtype_id.

## dimension\_object\_type\_table

Maps object type enumerations to text. Object types are used in many fact tables.



Column name	Data type	Description
type_id	int	Enumeration of object type.
type_name	varchar(32)	Text representation of type_id.
type_description_text	varchar(80)	Descriptive text for type_id.

## dimension\_publication\_type\_table

Maps publication enumerations to text. Publication enumerations are mostly used in health tables identifying whether the data is baseline (done on a schedule), incremental, or caused by a manual invocation of a health check.

Column name	Data type	Description
type_id	int	Enumeration of object type.
type_name	varchar(32)	Text representation of type_id.
type_description_text	varchar(80)	Descriptive text for type_id.

## dimension\_score\_table

Maps score enumerations to text. Scores are used in health tables to represent health as green (OK), yellow (warning), or red (error).

Column name	Data type	Description
score_id	int	Enumeration of object score.
score_name	varchar(32)	Text representation of score_id.
score_description_text	varchar(80)	Descriptive text for score_id.

## dimension\_severity\_table

Maps severity enumerations to text. Scores are used in event and health tables to represent severity from debug to emergency.



Column name	Data type	Description
severity_id	int	Enumeration of object severity.
severity_name	varchar(32)	Text representation of severity_id.
severity_description_text	varchar(80)	Descriptive text for severity_id.

## dimension\_statetype\_table

Maps severity enumerations to text. Scores are used in health tables to represent state from unknown to stopped.

Column name	Data type	Description
state_id	int	Enumeration of object state.
state_name	varchar(32)	Text representation of state_id.
state_description_text	varchar(80)	Descriptive text for state_id.

## dimension\_subhealthtype\_table

Maps subhealth enumerations to text. Subhealth enumerations are used in health tables to further clarify a health enumeration.

Column name	Data type	Description
subhealth_id	int	Enumeration of object subhealth.
subhealth_name	varchar(32)	Text representation of subhealth_id.
subhealth_description_text	varchar(80)	Descriptive text for subhealth_id.

## dimension\_substatetype\_table

Maps substate enumerations to text. Subhealth enumerations are used in health tables to further clarify a state enumeration.

Column name	Data type	Description
substate_id	int	Enumeration of object substate.
substate_name	varchar(32)	Text representation of substate_id.
substate_description_text	varchar(80)	Descriptive text for substate_id.



## dimension\_subtype\_type\_table

Associates object subtypes to object types to map object relationships.

Column name	Data type	Description
subtype_id	int	Enumeration of object subtype.
type_id	int	Enumeration of object type.

## dimension\_user\_association\_table

Used by charge-back reports to map Linux users to business groups and business functions. See *NSDA User Guide* for information on how to populate and modify the content of this table and how to use charge-back reports.

Column name	Data type	Description
association_id	int	Unique ID
group_longname	varchar(128)	Textual representation of the business group.
function_longname	varchar(128)	Textual representation of the business function.
user_longname	varchar(128)	Linux user name.



# Event tables

The Event tables provide access to centralized log messages as well as preindexed log messages.

See **Header columns** for a description of the header columns in the event tables. The following subsections describe the data columns in the event tables.

## event\_dsa\_table

Provides preindexed events for errors generated by the data service adapter (DSA); that is, the NSDA SQL Writers.

Column name	Data type	Description
error_text128	varchar (128)	Descriptive text specific to the error.
eventtime_utc_ts	timestamp(6)	The time the event was generated in Universal Co-ordinate Time (UTC) format.
file_column_num	Int	Column number at which an error was encountered. For errors related to interpreting files, such as configuration files.
file_line_num	Int	Line number at which an error was encountered. For errors related to interpreting files, such as configuration files.
file_name1024	varchar(1024)	Name of the file at which an error was encountered. For errors related to interpreting files, such as configuration files.
publication_column_full_name128	varchar(128)	Full field name in the publication related to the error. For errors related to publications.
publication_column_name128	varchar(128)	Field name in the publication related to the error. For errors related to publications.
publication_column2_name128	varchar(128)	Second field number to uniquely identify the error, if needed, for errors related to publications.
publication_data_type	Int	Data type related to the error. For errors related to publications.

*Table Continued*



Column name	Data type	Description
publication_name128	varchar(128)	Name of the publication related to the error. For errors related to publications.
repository_column_name128	varchar(128)	Repository column related to the error. For errors related to publications.
repository_table_name128	varchar(128)	Repository table related to the error. For errors related to publications.
routing_key_name256	varchar(256)	Routing key related to the error. For errors related to publications.
sql_error_code	Int	Numeric SQL error code, for SQL errors encountered.
sql_error_text1024	varchar(1024)	SQL error text, for SQL errors encountered.

## event\_text\_table

Contains textual representation of events and log messages.

Column name	Data type	Description
event_longtext	varchar(800)	The text in the event or log message.
tokenized_event_table_longname	varchar(128)	The name of the table that contains preindexed columns for the event or log message. NULL if no such table exists.



# Health tables

Health tables provide information generated by the NSDA health checks, which run on a periodic basis on the Data Service Instance.

See **Header columns** for a description of the header columns in the health tables. The following subsections describe the data columns in the health tables.

## health\_change\_table

Tracks changes in health and state for all monitored objects.

Column name	Data type	Description
health_detailed_table	varchar(128)	Health table containing detailed information about the health issue that caused the alert to be generated.
host_name	varchar(128)	The name of the host where the change originated.
change_type_enum	timestamp(6)	Enumeration of change type.

## health\_overview\_table

This view provides historical high-level instance health information. There are no data columns in this table, just health header columns described in **Header columns**.

## health\_summary\_table

Provides historical high-level instance health information for each monitored object in the instance.

Column name	Data type	Description
layer_name	varchar(32)	A textual description of the layer to which the subject belongs.
layer_current_score_enum	Int	An enumeration of the current score for the layer as follows: <ul style="list-style-type: none"><li>green(0)</li><li>yellow(1)</li><li>red (2)</li></ul>
layer_current_score_pct	double precision	Percentage of the current score for the layer.

*Table Continued*



Column name	Data type	Description
layer_previous_score_enum	Int	An enumeration of the current score for the layer as follows: <ul style="list-style-type: none"> <li>• green(0)</li> <li>• yellow(1)</li> <li>• red (2)</li> </ul>
layer_previous_score_pct	double precision	Percentage of the current score for the layer.
layer_score_change_utc_ts	timestamp(6)	The time a score change was determined in UTC format.
subject_name	varchar(32)	A textual description of the subject to which the object belongs.
subject_current_score_enum	Int	An enumeration of the current score for the subject as follows: <ul style="list-style-type: none"> <li>• green(0)</li> <li>• yellow(1)</li> <li>• red (2)</li> </ul>
subject_current_score_pct	double precision	Percentage of the current score for the subject.
subject_previous_score_enum	Int	An enumeration of the current score for the subject as follows: <ul style="list-style-type: none"> <li>• green(0)</li> <li>• yellow(1)</li> <li>• red (2)</li> </ul>
subject_previous_score_pct	double precision	Percentage of the current score for the subject.
subject_score_change_utc_ts	timestamp(6)	The time a score change was determined in UTC format.



# Metric tables

Metric views provide performance metrics information collected by the NSDA data sensors.

See **Header columns** for a description of the header columns in the metric tables. The following subsections describe the data columns in the metric tables.

## metric\_diskio\_summary\_table

Contains short-term aggregated avg, min, and max bandwidth usage for disks.

Column name	Data type	Description
util_avg_pct	double precision	The average disk %util for the instance.
node_min_longname	varchar(128)	The node with the disk that is the least used.
disk_min_name	varchar(32)	Disk with the least utilization.
util_min_pct	double precision	The lowest disk utilization in the instance.
node_max_longname	varchar(128)	The node with the disk that is the most used.
disk_max_name	varchar(32)	Disk with the most utilization.
util_max_pct	double precision	The highest disk utilization in the instance.

## metric\_diskio\_table

Provides summary disk utilization number and min/max disks based on disk utilization.

The data represents historical avg, min, and max bandwidth usage for disks.

Column name	Data type	Description
node_min_longname	varchar(128)	Host name to which the disks are associated.
util_avg_pct	double precision	The average disk %util for the instance.
disk_min_name	varchar(32)	Disk with the least utilization.
util_min_pct	double precision	The lowest disk utilization in the instance.

*Table Continued*



Column name	Data type	Description
disk_max_name	varchar(32)	Disk with the most utilization.
util_max_pct	double precision	The highest disk utilization in the node.

## metric\_heat\_summary\_table

Provides short-term information about the Heatmap metrics and their score value for the instance. The heat\_pct value is always 00.00-100.00 where 00.00% represents green state, 50% represents yellow state, and 100% represents red state. The values between green/yellow and yellow/red represent indicators for what hue to use.

Column name	Data type	Description
access_pct	double precision	Heatmap value for the access layer overall health.
dataservice_pct	double precision	Heatmap value for the data service layer overall health.
diskio_pct	double precision	Heatmap value for average disk I/O utilization.
infrastructure_pct	double precision	Heatmap value for the data-service infrastructure layer overall health
memory_pct	double precision	Heatmap value for average memory utilization.
networkio_pct	double precision	Heatmap value for average network I/O utilization.
node_pct	double precision	Heatmap value for average node utilization.
os_pct	double precision	Heatmap value for the OS layer overall health
server_pct	double precision	Heatmap value for the server layer overall health
storage_pct	double precision	Heatmap value for the storage layer overall health.

## metric\_jobquery\_summary\_table

Provides short-term high-level counts of job/queries by status.



Column name	Data type	Description
running_count	Int	Jobs/queries with a RUNNING status.
queued_count	Int	Jobs/queries with a QUEUED status. Not yet supported in SQL/MX.
completed_count	Int	Jobs/queries with a COMPLETED status.
failed_count	Int	Jobs/queries with a FAILED status. Not yet supported in SQL/MX.
holding_count	Int	Jobs/queries with a HOLDING status. Not yet supported in SQL/MX.
session_id	varchar(103)	The unique session ID generated by the MXCS SQL server at the time the connection is established.
mxcs_service_name	varchar(6)	Name of the MXCS service.
process_name	varchar(32)	The process name of the server (MXOSRVR).
datasource_name	varchar(128)	Datasource name.
user_longname	varchar(50)	User submitting work.

## metric\_jobquery\_table

Provides high-level information about jobs/queries.

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
job_type	Varchar(32)	Reserved for future use.
cpu_us	largeint	Number of microseconds the job/query used the CPU.
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_lct_ts	timestamp(6)	Time job ended in Local Civil Time. Null if running.

*Table Continued*



Column name	Data type	Description
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
memory_bytes	largeint	Amount of memory (in bytes) used by the job/query.
read_bytes	largeint	Number of bytes read by the job/query.
running_bool	largeint	Flag indicating whether the job is running.
runtime_us	largeint	Time (in microseconds) job ran/has been running.
starttime_lct_ts	timestamp(6)	Time job started in Local Civil Time.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.
written_bytes	largeint	Number of bytes written by the job/query.
Row_count	Int	Number of rows accessed by the job/query.
Disk_ios	Int	Number of disk I/Os by the job/query.

## metric\_jobquery\_topncputime\_table

Provides the top-n resource consuming jobs/queries based on CPU time.

A join on transaction\_id exposes counters bytes read from cache/disk, memory allocated, written rows, and so on. The information is per operator, which means that sums are calculated.

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
cpu_us	largeint	Number of microseconds the job/query used the CPU.

*Table Continued*



Column name	Data type	Description
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.
running_bool	largeint	Flag indicating whether the job is running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.

## metric\_jobquery\_topndiskios\_table

Provides the top-n resource consuming jobs/queries based on disk I/Os.

For NonStop SQL/MX, the information comes from the job data:

- user-name
- job-id, start/end time
- job-status
- cpu-ms
- physical-memory
- hdfs-read-bytes
- hdfs-written-bytes

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
disk_ios	largeint	Number of disk I/Os by the job/query.
details_table_longname	varchar(128)	Name of table where job/query details can be found.

*Table Continued*



Column name	Data type	Description
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.
running_bool	largeint	Flag indicating whether the job is running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.

## metric\_jobquery\_topmemory\_table

This view provides the top-n resource consuming jobs/queries based on memory usage.

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.
memory_bytes	largeint	Amount of memory (in bytes) used by the job/query.
running_bool	largeint	Flag indicating whether the job is running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.



## metric\_jobquery\_topnreads\_table

This view provides the top-n resource consuming jobs/queries based on read I/O bytes.

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.
read_bytes	largeint	Number of bytes read by the job/query.
running_bool	largeint	Flag indicating whether the job is running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.

## metric\_topnrows\_table

This view provides the top-n resource consuming jobs/queries based on table rows accessed.

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.

*Table Continued*



Column name	Data type	Description
row_count	largeint	Number of rows accessed from the tables referenced by the query. This number includes the records examined by the disk process, the file system, and the executor.
running_bool	largeint	Flag indicating whether the job is running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.

## metric\_jobquery\_topnrows\_table

Column name	Data type	Description
user_longname	varchar(128) not null	User submitting work.
jobquery_longname	varchar (160) not null	Job/Query ID.
status_name	varchar (32) not null	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
starttime_lct_ts	timestamp (6) not null	Time job started in LCT.
starttime_utc_ts	timestamp (6) not null	Time job started in UTC.
endtime_lct_ts	timestamp (6) default null	Time job ended in LCT. NULL if running.
endtime_utc_ts	timestamp (6) default null	Time job ended in UTC. NULL if running.
running_bool	largeint default null	Flag indicating whether the job is running.

*Table Continued*



Column name	Data type	Description
row_count	largeint default null	Number of rows accessed from the tables referenced by the query. This number includes the records examined by the disk process, the file system, and the executor.
details_table_longname	varchar(128) default null	Name of table where job/query details can be found.

## metric\_jobquery\_topnruntime\_table

This view provides the top-n resource consuming jobs/queries based on the timestamp of the latest reported record minus the start time of the job/query.

Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.
running_bool	largeint	Flag indicating whether the job is running.
runtime_us	largeint	Time (in microseconds) job ran/has been running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.

## metric\_jobquery\_topnwrites\_table

This view provides the top-n resource consuming jobs/queries based on the write I/O bytes.



Column name	Data type	Description
jobquery_longname	varchar(160)	Job/query ID.
details_table_longname	varchar(128)	Name of table where job/query details can be found.
endtime_utc_ts	timestamp(6)	Time job ended in UTC. Null if running.
endtime_lct_ts	timestamp(6)	Time job ended in LCT. Null if running.
running_bool	largeint	Flag indicating whether the job is running.
starttime_utc_ts	timestamp(6)	Time job started in UTC.
starttime_lct_ts	timestamp(6)	Time job started in LCT.
status_name	varchar(32)	RUNNING, COMPLETED, QUEUED, FAILED, and so on.
user_longname	varchar(128)	User submitting work.
written_bytes	largeint	Number of bytes written by the job/query.

## metric\_loadqueue\_summary\_table

This view provides ongoing avg, min, and max task queue information (the number of active and waiting tasks on each node) for the entire instance in form of a percentage used for heatmap-oriented charting.

Column name	Data type	Description
load_avg_stat	double precision	The average number of running tasks the instance.
loadqueue_avg_pct	double precision	The average load-queue percent the instance.
node_min_longname	varchar(128)	The node with the smallest average number of running tasks.
load_min_stat	double precision	The number of average number of tasks for min_node.
loadqueue_min_pct	double precision	The node with the smallest load-queue percent in the instance.

*Table Continued*



Column name	Data type	Description
node_max_longname	varchar(128)	The node with the largest average number of running tasks.
load_max_stat	double precision	The number of average number of tasks for max_node.
loadqueue_max_pct	double precision	The node with the most load-queue percent in the instance.

## metric\_loadqueue\_table

This view provides ongoing avg, min, and max task queue information (the number of active and waiting tasks on each node) for a node on the instance.

Column name	Data type	Description
load_avgmin_stat	double precision	The average number of running tasks on the node.
loadqueuemin_avg_pct	double precision	The average load-queue percent on the node.
load_avg_5min_stat	double precision	The average number of running tasks on the node.
loadqueue_5min_avg_pct	double precision	The average load-queue percent on the node.
load_avg5min_stat	double precision	The average number of running tasks on the node.
loadqueue5min_avg_pct	double precision	The average load-queue percent on the node.

## metric\_memory\_summary\_table

This view provides short-term avg., min., and max. memory, and swap usage information (in pct) for the entire instance.



Column name	Data type	Description
swap_avg_pct	double precision	The average memory swap space in use (0.00 - 100.00)
swap_max_pct	double precision	The max. memory swap space in use (0.00 - 100.00)
swap_min_pct	double precision	The min. memory swap space in use (0.00 - 100.00)
swap_node_max_longname	varchar(128)	The node with the most swap space in use. If min_swap = 0.00 then this column is set to null.
swap_node_min_longname	varchar(128)	The node with the least swap space in use. If min_swap = 0.00 then this column is set to null.
used_avg_pct	double precision	The average pct memory in use (0.00 - 100.00)
used_max_pct	double precision	The max. pct memory in use (0.00 - 100.00)
used_min_pct	double precision	The min. pct memory in use (0.00 - 100.00)
used_node_max_longname	varchar(128)	The node with the most memory in use.
used_node_min_longname	varchar(128)	The node with the least memory in use.

## metric\_memory\_table

This view provides memory metrics for each node.



Column name	Data type	Description
active_bytes	largeint	Memory, in bytes, that has been used recently and has not yet been reclaimed. This memory is most likely to be used in the next interval. It will not be swapped to disk unless there is a priority requirement to do so.
cached_bytes	largeint	In-memory cache for files read from the disk (that is, the page cache), in bytes. Does not include wap_cached_bytes.
free_memory_bytes	largeint	Free memory, in bytes.  This value is obtained by adding the unused physical memory in the higher address region (Linux "HighFree") and the unused physical memory in the lower address region (Linux "LowFree").  NOTE: While this value reflects all of the memory, "high" memory and "low" memory have different uses. User programs and page cache rely on "high" memory. The Linux kernel relies on "low" memory and requires sufficient amounts of available "low" memory to perform its functions.
inactive_bytes	largeint	Memory, in bytes, that has been used less recently than ACTIVE memory. It is eligible to be reclaimed for other purposes.
raw_disk_temp_storage_blocks_bytes	largeint	Temporary storage for raw disk blocks, in bytes. This must be around 20,480,000 bytes.
swap_cached_bytes	largeint	Memory, in bytes, that was swapped out, then in again. This memory remains in the swap file, allowing for faster swap I/O. If the memory is not needed for something else, the contents need not be written.
swap_space_free_bytes	largeint	Memory, in bytes, that has been moved from AM and is temporarily available on the disk.

*Table Continued*



Column name	Data type	Description
swap_space_total_bytes	largeint	Total amount of swap space available, in bytes.
total_memory_bytes	largeint	Total usable memory, in bytes. (Physical memory minus a few reserved bits and the Linux kernel binary node.)

## metric\_network\_summary\_table

This view provides short-term avg, min, and max memory and swap usage information (in pct) for the entire instance.

Column name	Data type	Description
rx_avg_bytes	int	Average receive in bytes.
rx_avg_pct	double precision	Average receive in percentage.
rx_if_max_longname	varchar(128)	Name of interface with the most receive processing in interval.
rx_if_min_longname	varchar(128)	Name of interface with the least receive processing in interval.
rx_max_bytes	int	Max receive in bytes.
rx_max_pct	double precision	Max receive in percentage.
rx_min_bytes	int	Min receive in bytes.
rx_min_pct	double precision	Min receive in percentage.
rx_node_max_longname	varchar(128)	Name of node with the most receive processing in interval.
rx_node_min_longname	varchar(128)	Name of node with the least receive processing in interval.
tx_avg_bytes	int	Average transmit in bytes.
tx_avg_pct	double precision	Average transmit in percentage.
tx_if_max_longname	varchar(128)	Name of interface with the most transmit processing in interval.

*Table Continued*



Column name	Data type	Description
tx_if_min_longname	varchar(128)	Name of interface with the least transmit processing in interval.
tx_max_bytes	int	Max transmit in bytes.
tx_max_pct	double precision	Max transmit in percentage.
tx_min_bytes	int	Min transmit in bytes.
tx_min_pct	double precision	Min transmit in percentage.
tx_node_max_longname	varchar(128)	Name of node with the most transmit processing in interval.
tx_node_min_longname	varchar(128)	Name of node with the least transmit processing in interval.
util_avg_pct	double precision	Average interface utilization in percentage.

## metric\_network\_table

This view provides Network metrics for each node. The information is based on `/proc/net/dev`.

Column name	Data type	Description
interface_longname	varchar(128)	Identifier for the logical network interface as known by the Linux kernel.
receive_bytes	largeint	Number of total bytes of data received since this device has started.
receive_packets_count	largeint	Number of total packets of data received since this service was started.
receive_errors_count	largeint	Total number of receive errors detected by the device driver.
receive_packets_drops_count	largeint	Total number of packets dropped by the device driver when receiving data.
receive_fifo_errors_count	largeint	Number of FIFO buffer errors when receiving data.

*Table Continued*



Column name	Data type	Description
receive_fifo_framing_errors_count	largeint	Number of FIFO packet framing errors when receiving data.
receive_packets_compressed_count	largeint	Number of compressed packets received by the device driver since the device was started.
receive_packets_multicast_frames_count	largeint	Number of multicast frames received by the device driver.
transmit_bytes	largeint	Number of total bytes of data transmitted since this device was started.
transmit_packets_count	largeint	Number of total packets of data transmitted since this device was started.
transmit_errors_count	largeint	Total number of transmit errors detected by the device driver.
transmit_packets_drops_count	largeint	Total number of packets dropped by the device driver when transmitting data.
transmit_fifo_errors_count	largeint	Number of FIFO buffer errors when transmitting data.
transmit_collision_errors_count	largeint	Number of collisions detected on the interface when transmitting data.
transmit_carrier_losses_count	largeint	Number of carrier losses detected by the device driver when transmitting data.
transmit_packets_compressed_count	largeint	Number of compressed packets transmitted by the device driver since device was started.

## metric\_nonstop\_query\_plan\_table

This table provides information about a NonStop query plan.



Column name	Data type	Description
cpu	int	The CPU in which the query ran.
plan_id	largeint	A unique identifier for the plan. Generated by the compilerQuery plan ID number, as returned by RTS.
plan_length	Int	The length in bytes of the base64 decoded plan data. The length is the total length of the entire plan data as returned by RTS.
plan_data	varchar(5462)	BASE64-Encoded plan data from RTS.
total_seq	Int	Total number of sequences – a value greater than one indicates if there are more data records for this plan, and if so, how many. This typically happens when the plan data must be broken down into multiple rows when the data does not fit into one row. The PLAN_LENGTH of the first row will contain the total plan data length and for any subsequent rows it will be zero.
current_seq	Int	Indicates the current row in the sequence. The sequence starts from 1.

## metric\_nonstop\_query\_progress\_table

This table provides information about a NonStop query by retrieving data from WMS/RTS.



Column name	Data type	Description
jobquery_longname	varchar(160)	Job Identifier. Jobquery_longname provides a simple way to find a unique query when looking for query details.
status_name	varchar(9)	Status of the query.  Actual values: <ul style="list-style-type: none"> <li>• WAITING<sup>1</sup></li> <li>• EXECUTING</li> <li>• HOLDING <sup>1</sup></li> <li>• REJECTED <sup>1</sup></li> <li>• COMPLETED</li> <li>• UNKNOWN</li> </ul> Normalized: <ul style="list-style-type: none"> <li>• RUNNING</li> <li>• COMPLETED</li> <li>• QUEUED <sup>1</sup></li> <li>• FAILED <sup>1</sup></li> <li>• UNKNOWN</li> </ul>
substatus_name	varchar(30)	Substatus of the query.
starttime_utc_ts	timestamp(6)	The time the query started in Universal Co-ordinate Time (UTC) format.
starttime_lct_ts	timestamp(6)	The time the query started in Local Civil Time (LCT) format.
endtime_utc_ts	timestamp(6)	The time the query ended in Universal Co-ordinate Time (UTC) format. Null if query is active.
endtime_lct_ts	timestamp(6)	The time the query ended in Local Civil Time (LCT) format. Null if query is active.
running_bool	Int	Indicates whether the query is running.
long_service_name	varchar(96)	Name of the service.
estimated_cost	double precision	Estimated cost associated with executing the query.

*Table Continued*



<b>Column name</b>	<b>Data type</b>	<b>Description</b>
estimated_cpu_time_secs	double precision	Estimated seconds of processor time to execute query execution plan instructions.
estimated_io_time_secs	double precision	Estimated seconds of I/O time (seek plus data transfer) to execute the query.
estimated_msg_time_secs	double precision	Estimated seconds for the messaging for the query.
estimated_idle_time_secs	double precision	Estimated seconds to wait for events to occur.
estimated_total_time_secs	double precision	Estimated seconds to execute the query.
estimated_row_count	double precision	Estimated number of rows the query will return.
estimated_total_memory_kb	double precision	Estimated total memory per CPU used by query plan, in kilobytes.
estimated_resource_usage_secs	Int	Estimated amount of work to be done by the CPUs for the query, in seconds.
cmp_affinity_num	Int	The choice of CPU subsets to be used for placement of the Executor Server Process (ESP).
cmp_dop	Int	The degree of parallelism used by the query.
cmp_txn_needed	Int	Denotes whether the query is required to execute under a transaction.
cmp_mandatory_x_prod	Int	This counter flags queries that have joins without predicates.
cmp_missing_stats	Int	Denotes if single-column missing statistics warnings were raised during compilation (with SQLCODE 6008 or 6011).
cmp_joins_count	Int	The number of joins in the query plan.
cmp_full_scan_on_table	Int	Denotes if a full scan will be performed on at least one table by the query plan.
cmp_high_dp2_max_buf_usage	Int	For future use.

*Table Continued*



Column name	Data type	Description
cmp_rows_accessed_full_scan_count	double precision	The estimated number of rows that will be accessed by a full scan of a table. This counter is valid only if CMP_FULL_SCAN_ON_TABLE is set.
cmp_dp2_rows_accessed_count	double precision	The estimated number of rows that will be accessed by all the scan, insert, update, and delete operators in the query plan.
cmp_dp2_rows_used_count	double precision	The estimated number of rows that will be accessed by scan and insert operators in the query plan.
process_name	varchar(32)	The process name of the server (MXOSRVR).
query_name	varchar(200)	Application name and computer name delimited by the    character.
application_name	varchar(200)	Application name.
computer_name	varchar(200)	Computer name
user_longname	varchar(50)	User name.
datasource_name	varchar(128)	Datasource name.
tenant_db_name	varchar(128)	Tenant database name the query belongs to.
starttime_lct_text	varchar(26)	Local civil time when the query was registered in WMS. YYYY-MM-DD HH:MM:SS.aaabbb (aaa = ms, bbb = microseconds).
entrytime_lct_text	varchar(26)	Local civil time when the query was inserted into the service. YYYY-MM-DD HH:MM:SS.aaabbb (aaa = ms, bbb = microseconds).
entrytime_jts	largeint	Unformatted local civil time when the query was inserted into the service.
statement_id	varchar(160)	Statement identifier specified by the client application.
statement_type	varchar(21)	Statement type.
cmp_starttime	largeint	Compilation start time.
cmp_starttime_utc_ts	timestamp(6)	Compilation start time in Universal Co-ordinate Time (UTC) format.
cmp_elapsedtime	largeint	Compilation elapsed time.

*Table Continued*



<b>Column name</b>	<b>Data type</b>	<b>Description</b>
cmp_endtime	largeint	Compilation end time.
cmp_endtime_utc_ts	timestamp(6)	Compilation end time in Universal Co-ordinate Time (UTC) format. Null if query is compiling.
exec_starttime	largeint	Execution start time.
exec_starttime_utc_ts	timestamp(6)	Execution start time in Universal Co-ordinate Time (UTC) format.
exec_endtime	largeint	Execution end time.
exec_endtime_utc_ts	timestamp(6)	Execution end time in Universal Co-ordinate Time (UTC) format. Null if query is active.
exec_state	varchar(21)	Execution state.
runtime_us	largeint	Elapsed time of query execution in microseconds.
waittime_us	largeint	Length of time spent in WAITING state in microseconds.
holdtime_us	largeint	Length of time spent in HOLD state in microseconds.
accessed_row_count	largeint	Number of rows accessed from the tables referenced by the query. This number includes the records examined by the disk process, the file system, and the executor.
used_row_count	largeint	Number of rows used by the query.
message_count	largeint	Number of messages sent to execute operations on the table referenced by the query.
message_bytes	largeint	Number of bytes to send and receive messages for this query.
stats_bytes	largeint	Number of bytes to send statistics messages for this query.
disk_ios	largeint	Number of disk reads caused by accessing the tables referenced by the query.
lock_waits	largeint	Number of times the query waited for a lock request.

*Table Continued*



<b>Column name</b>	<b>Data type</b>	<b>Description</b>
lock_escalations	largeint	Number of times a row lock was escalated to a file lock for the query.
cpu_us	largeint	Actual CPU time in microseconds spent by all DP2 processes involved in executing the query.
opens_count	largeint	Number of OPEN calls performed by the executor on behalf of this process.
opentime	largeint	Time the process spent doing OPENS.
sql_error_code	Int	Top-level error code returned by the query, indicating whether the query completed successfully or with warnings or errors.
affected_rows	largeint	Number of rows inserted, updated, or deleted by the query.
stats_error_code	Int	Warning number returned to the statistics collector while it is obtaining statistics from runtime statistics.
sql_space_alloc_kb	Int	Amount, in kilobytes, of the space type of memory allocated in the master and Executor Server Processes (ESPs).
sql_space_used_kb	Int	Amount, in kilobytes, of the space type of memory used in the master and ESPs involved in processing the query.
sql_heap_alloc_kb	Int	Amount, in kilobytes, of the heap type of memory allocated in the master and ESPs involved in processing the query.
sql_heap_used_kb	Int	Amount, in kilobytes, of the heap type of memory used in the master and ESPs involved in processing the query.
sql_cpu_time_us	largeint	Approximation, in microseconds, of the total CPU time spent in the master and ESPs involved in processing the query.
eid_space_alloc_kb	Int	Amount, in kilobytes, of the space type of memory allocated in the EID (executor code in DP2 processes) involved in processing the query.
eid_space_used_kb	Int	Amount, in kilobytes, of the space type of memory used in the EID (executor code in DP2 processes) involved in processing the query.

*Table Continued*



<b>Column name</b>	<b>Data type</b>	<b>Description</b>
eid_heap_alloc_kb	Int	Amount, in kilobytes, of the heap type of memory allocated in the EID (executor code in DP2 processes) involved in processing the query.
eid_heap_used_kb	Int	Amount, in kilobytes, of the heap type of memory used in the EID (executor code in DP2 processes) involved in processing the query.
total_mem_alloc_kb	Int	The sum of SQL_SPACE_USED and SQL_HEAP_USED.
total_mem_used_kb	Int	The maximum memory used by a query.
est_accessed_row_count	double precision	Estimated number of rows accessed by the query.
est_used_row_count	double precision	Estimated number of rows used by the query.
process_created_count	largeint	Number of processes generated by the executor on behalf of this query.
process_create_time	largeint	Time spent creating processes.
query_text	varchar(255)	The SQL text for the query. This contains only the first 255 characters of the SQL text.
query_text_length	Int	Length of query text in bytes.
cmp_lastupdated_lct	largeint	Local Civil Time (LCT) when the shared memory block was last updated during compilation.
exec_lastupdated_lct	largeint	Local Civil Time (LCT) when the shared memory block was last updated during execution.
entrytime_utc_ts	timestamp(6)	Time when the query was inserted into the service in Universal Co-ordinate Time (UTC) format.
cmp_lastupdated_utc_ts	timestamp(6)	Time when the shared memory block was last updated in during compile in Universal Co-ordinate Time (UTC) format.
exec_lastupdated_utc_ts	timestamp(6)	Time when the shared memory block was last updated in during execution in Universal Co-ordinate Time (UTC) format.

*Table Continued*



Column name	Data type	Description
query_text_id	largeint	Query text ID.
plan_id	largeint	Query plan ID.
mxcs_service_name	varchar(6)	Name of the MXCS service.
cpu	Int	CPU number.

<sup>1</sup> Not yet supported in SQL/MX

## metric\_nonstop\_query\_text\_table

Column name	Data type	Description
text_id	largeint	Query Text identifier which is unique for the CPU.
text_length	Integer not null	Total length of the entire query text. See TOTAL_SEQ for more info.
text_data	varchar (4096) not null	Text for the SQL query syntax.
total_seq	Integer not null	A value greater than 1 indicates if there are more data records for this query text and if so how many. This typically happens when the query text needs to be broken down into multiple rows when the data does not fit into one row. The text_length of the first row will contain the total query text length and for any subsequent rows it will be zero.
current_seq	Integer not null	Indicates the current row in the sequence. The sequence starts from 1.
cpu	Integer not null	CPU number where the query originated.



## metric\_nonstop\_session\_table

Column name	Data type	Description
session_id	varchar (103) not null	The unique session ID generated by the MXCS SQL server at the time the connection is established.
State	varchar (9) not null	State of the session: OPEN, CLOSED or UNKNOWN.
application_name	varchar (120) not null	The client application name or the main caption.
computer_name	varchar (64) not null	The client workstation TCP/IP name or the NetBios/network name.
user_name	varchar (128) not null	The Guardian user name or the External Username for a tenant user.
server_name	varchar (32) not null	The process name of the NonStop server (MXCS SQL server).
mxcs_service_name	varchar (6) not null	The MXCS service from which the connection was established.
wms_service_name	varchar (96) not null	The name of WMS Service associated with the session. The default name is HP_DEFAULT_SERVICE.
data_source_name	varchar (128) not null	The client data source name used at the time of connection.
tenant_db_name	varchar (128) not null	The tenant database name to which the session belongs in a NonStop multi-tenancy environment

## metric\_nonstop\_tenant\_access\_table

Column name	Data type	Description
user_id	integer not null	The Guardian user ID. A known user to the system as grantor, grantee, or owner.  -1 if the user is PUBLIC.  -2 if the user is SYSTEM.



## metric\_nonstop\_tenant\_table

Column name	Data type	Description
tenant_name	varchar (128) not null	The Tenant database name that corresponds to the tenant_id field in message header.

## metric\_nonstop\_user\_table

Column name	Data type	Description
user_id	integer not null	The Guardian user ID. A known user to the system as grantor, grantee, or owner.  -1 if the user is PUBLIC.  -2 if the user is SYSTEM.
user_name	varchar (32) not null	The External Username for a Guardian user identified by user_id field on NonStop SQL/MX.
guardian_name	varchar (18) not null	The corresponding Guardian username of the user identified by the user_id field.

## metric\_physdiskstats\_table

Provides Linux-level physical disk metrics. The data is based on `/proc`, unmanipulated.

These column values are direct from Linux virtual file system values. For guidance in how to use these fields, see the appropriate Linux documentation.

Column name	Data type	Description
calendar_lct_date	date	—
calendar_utc_date	date	—
component_id	integer unsigned	—
gen_utc_ts	timestamp (6)	—
group_id	integer unsigned	Not used.

*Table Continued*



Column name	Data type	Description
host_id	integer unsigned	—
instance_id	integer unsigned	—
interface_name	varchar (32)	Device name
io_in_progress_count	largeint	Not used. Number of I/Os currently in progress. Incremented as requests are given to appropriate struct and decremented as they are finished. Can go to zero.
io_time_ms	largeint	Not used. Number of ms spent doing I/Os. Increases as num_ios_currently_in_progress is nonzero.
ip_address	varchar (32)	—
node_id	integer unsigned	—
pnid_id	integer unsigned	—
process_id	int	—
queue_avg_size	double precision	The average queue length of the requests that were issued to the device.
read_completed_count	largeint	Not used. Total number of reads completed successfully since last boot.
read_kb_persec	double precision	The kilo bytes read from the device per second.
read_merged_count	largeint	Not used. Num reads merged. Reads adjacent to each other might be merged for efficiency. This is the count of times that it happened since last boot.
read_requests_merged_per_second_count	int	The number of read requests merged per second that were queued to the device.

*Table Continued*



Column name	Data type	Description
read_requests_per_second_count	largeint	The number of read requests that were issued to the device per second.
read_sectors_count	largeint	Not used. Total number of sectors read successfully since last boot.
read_time_ms	largeint	Not used. Total number of ms spent by all reads as measured from <code>__make_request()</code> to <code>end_that_request_last()</code> .
request_sector_avg_size	double precision	The average size (in sectors) of the requests that were issued to the device.
se_name	varchar (32)	Not used.
sequence_num	integer unsigned	—
service_avg_time_ms	large int	The average service time (in ms) for I/O requests that were issued to the device. Warning! Do not trust this field anymore. This field will be removed in a future sysstat version.
service_name	varchar (32)	—
tenant_id	int	Not used.
thread_id	integer unsigned	—
time_interval_ms	largeint	Time interval between two samples (in ms).
upsert_lct_ts	timestamp (6)	—
upsert_utc_ts	timestamp (6)	—
util_pct	double precision	Percentage of CPU time during which I/O requests were issued to the device (bandwidth utilization for the device). Device saturation occurs when this value is close to 100%.

*Table Continued*



Column name	Data type	Description
wait_avg_time_ms	largeint	The average time (in ms) for I/O requests issued to the device to be served. This includes the time spent by the requests in queue and the time spent servicing them.
weighted_time_doing_ios_ms	largeint	Not used. Weighted ms doing I/Os. Incremented at each I/O Start, I/O completion, I/O merge, or read of these stats, by the number of I/Os in progress times the number of ms doing I/O since the last update of this field. Measures I/O completion time and backlog accumulating.
write_completed_count	largeint	Not used. Number of writes completed successfully since last boot
write_kb_persec	double precision	The kilo bytes written to the device per second.
write_merged_count	largeint	Num writes merged writes adjacent to each other might be merged for efficiency. This is the count of times that it happened since last boot.
write_requests_merged_per_second_count	Int	The number of write requests merged per second that were queued to the device.
write_requests_per_second_count	largeint	The number of write requests that were issued to the device per second.
write_sectors_count	largeint	Total number of sectors written since last boot
write_time_ms	largeint	Total number of ms spent by all writes as measured from __make_request() to end_that_request_last()



# Support and other resources

## Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:  
<https://www.hpe.com/info/assistance>
- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:  
<https://www.hpe.com/support/hpesc>

### Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

## Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

### Hewlett Packard Enterprise Support Center

<https://www.hpe.com/support/hpesc>

### Hewlett Packard Enterprise Support Center: Software downloads

<https://www.hpe.com/support/downloads>

### My HPE Software Center

<https://www.hpe.com/software/hpesoftwarecenter>

- To subscribe to eNewsletters and alerts:  
<https://www.hpe.com/support/e-updates>
- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:  
<https://www.hpe.com/support/AccessToSupportMaterials>





---

**IMPORTANT:** Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

---

## Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

### Remote support and Proactive Care information

#### HPE Get Connected

<https://www.hpe.com/services/getconnected>

#### HPE Proactive Care services

<https://www.hpe.com/services/proactivecare>

#### HPE Datacenter Care services

<https://www.hpe.com/services/datacentercare>

#### HPE Proactive Care service: Supported products list

<https://www.hpe.com/services/proactivecaresupportedproducts>

#### HPE Proactive Care advanced service: Supported products list

<https://www.hpe.com/services/proactivecareadvancedsupportedproducts>

### Proactive Care customer information

#### Proactive Care central

<https://www.hpe.com/services/proactivecarecentral>

#### Proactive Care service activation

<https://www.hpe.com/services/proactivecarecentralgetstarted>

## Warranty information

To view the warranty information for your product, see the links provided below:

### HPE ProLiant and IA-32 Servers and Options

<https://www.hpe.com/support/ProLiantServers-Warranties>

### HPE Enterprise and Cloudline Servers

<https://www.hpe.com/support/EnterpriseServers-Warranties>

### HPE Storage Products

<https://www.hpe.com/support/Storage-Warranties>

### HPE Networking Products

<https://www.hpe.com/support/Networking-Warranties>

## Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

<https://www.hpe.com/support/Safety-Compliance-EnterpriseProducts>



### **Additional regulatory information**

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

**<https://www.hpe.com/info/reach>**

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

**<https://www.hpe.com/info/ecodata>**

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

**<https://www.hpe.com/info/environment>**

## **Documentation feedback**

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**[docsfeedback@hpe.com](mailto:docsfeedback@hpe.com)**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

