HP Neoview Workload Management Services Guide

HP Part Number: 544806-001
Published: April 2008
Edition: HP Neoview Release 2.3

Legal Notice

Confidential computer software. Valid license from HP 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.

The information contained herein is subject to change without notice. The only warranties for HP 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. HP shall not be liable for technical or editorial errors or omissions contained herein.

Export of the information contained in this publication may require authorization from the U.S. Department of Commerce.

Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation.

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

Java is a U.S. trademark of Sun Microsystems, Inc.

Motif, OSF/1, UNIX, X/Open, and the "X" device are registered trademarks, and IT DialTone and The Open Group are trademarks of The Open Group in the U.S. and other countries.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE OSF MATERIAL PROVIDED HEREIN, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
OSF shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.
© 1990, 1991, 1992, 1993 Open Software Foundation, Inc. The OSF documentation and the OSF software to which it relates are derived in part from materials supplied by the following:
© 1987, 1988, 1989 Carnegie-Mellon University. © 1989, 1990, 1991 Digital Equipment Corporation. © 1985, 1988, 1989, 1990 Encore Computer Corporation. © 1988 Free Software Foundation, Inc. © 1987, 1988, 1989, 1990, 1991 Hewlett-Packard Company. © 1985, 1987, 1988, 1989, 1990, 1991, 1992 International Business Machines Corporation. © 1988, 1989 Massachusetts Institute of Technology. © 1988, 1989, 1990 Mentat Inc. © 1988 Microsoft Corporation. © 1987, 1988, 1989, 1990, 1991, 1992 SecureWare, Inc. © 1990, 1991 Siemens Nixdorf Informationssysteme AG. © 1986, 1989, 1996, 1997 Sun Microsystems, Inc. © 1989, 1990, 1991 Transarc Corporation.
OSF software and documentation are based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. OSF acknowledges the following individuals and institutions for their role in its development: Kenneth C.R.C. Arnold, Gregory S. Couch, Conrad C. Huang, Ed James, Symmetric Computer Systems, Robert Elz. © 1980, 1981, 1982, 1983, 1985, 1986, 1987, 1988, 1989 Regents of the University of California.

Table of Contents

List of Figures

List of Tables

About This Document

This manual describes how to use the Neoview Workload Management Services (WMS) to perform workload and resource management on a Neoview data warehousing platform. WMS commands enable you to define service levels and resource thresholds, monitor system resources, and manage queries.

Intended Audience

This manual is intended for customers who are using WMS to manage workload on a Neoview data warehousing platform. If you are in HP Support, see additional information about WMS in a chapter of the *Neoview Database Support Guide*.

New and Changed Information in This Edition

This is a new manual for customers.

Document Organization

Chapter 1: Introducing Neoview Workload Management Services (WMS)	Introduces WMS and describes how it works.
Chapter 2: Getting Started	Describes how to install WMS and related products.
Chapter 3: Configuring WMS	Describes how to set up services, thresholds, and other settings so that you can manage workload using WMS.
Chapter 4: Associating Queries With Services	Describes how to associate queries with services.
Chapter 5: Managing Workload in WMS	Describes how to monitor and manage queries, services, and system resources using WMS.
Appendix A: WMS Commands	Provides syntax, considerations, and examples for the WMS commands.

Notation Conventions

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS

Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

SELECT

`Italic Letters`

Italic letters, regardless of font, indicate variable items that you supply. Items not enclosed in brackets are required. For example:

`file-name`

`Computer Type`

Computer type letters within text indicate case-sensitive keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

```
myfile.sh
```

`Bold Text`

Bold text in an example indicates user input typed at the terminal. For example:

```
ENTER RUN CODE

?123
CODE RECEIVED:      123.00
```

The user must press the Return key after typing the input.

[ ] Brackets

Brackets enclose optional syntax items. For example:

```
DATETIME [start-field TO] end-field
```

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

```
DROP SCHEMA schema [CASCADE]
                   [RESTRICT]

DROP SCHEMA schema [ CASCADE | RESTRICT ]
```

{ } Braces

Braces enclose required syntax items. For example:

```
FROM { grantee[, grantee]...}
```

A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
INTERVAL { start-field TO end-field }
         { single-field }

INTERVAL { start-field TO end-field  | single-field }
```

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
{expression | NULL}
```

… Ellipsis

An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
ATTRIBUTE[S] attribute [, attribute]...

{, sql-expression}...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
expression-n...
```

Punctuation

Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
DAY (datetime-expression)
```

```
@script-file
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"{" module-name [, module-name]... "}"
```

Item Spacing

Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
DAY (datetime-expression)

DAY(datetime-expression)
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
myfile.sh
```

Line Spacing

If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
match-value [NOT] LIKE pattern

   [ESCAPE esc-char-expression]
```

Related Documentation

This manual is part of the HP Neoview customer library.

Neoview Customer Library

- **Administration**

Neoview Character Sets Administrator's Guide	Information for database administrators and end users of the Neoview Character Sets product, including rules for defining and managing character data using SQL language elements, capabilities and limitations of Neoview client applications, troubleshooting character set-related problems, and enabling Pass-Through mode in the ISO88591 configuration.
Neoview Command Interface (NCI) Guide	Information about using the HP Neoview Command Interface to run SQL statements interactively or from script files.
Neoview Command Interface (NCI) Online Help	Command-line help that describes the commands supported in the current operating mode of Neoview Command Interface.
Neoview Database Administrator's Guide	Information about how to load and manage the Neoview database by using the Neoview DB Admin and other tools.
Neoview DB Admin Online Help	Context-sensitive help topics that describe how to use the HP Neoview DB Admin management interface.
Neoview Guide to Stored Procedures in Java	Information about how to use stored procedures that are written in Java within a Neoview database.

Neoview Management Dashboard Client Guide for Database Administrators	Information on using the Dashboard Client, including how to install the Client, start and configure the Client Server Gateway (CSG), use the Client windows and property sheets, interpret entity screen information, and use Command and Control to manage queries from the Client.
Neoview Management Dashboard Online Help	Context-sensitive help topics that describe how to use the Neoview Management Dashboard Client.
Neoview Repository User Guide	Information about using the Repository, including descriptions of Repository views and guidelines for writing Neoview SQL queries against the views.
Neoview Owner's Manual	Site-planning information and basic hardware information.
Neoview Performance Analyzer Online Help	Context-sensitive help topics that describe how to use the Neoview Performance Analyzer to analyze and troubleshoot query-related issues on the Neoview data warehousing platform.
Neoview Query Guide	Information about reviewing query execution plans and investigating query performance of Neoview databases.
Neoview Reports Online Help	Help topics that describe how to use the HP Neoview Reports Tool.
Neoview Transporter User Guide	Information about processes and commands for loading data into your Neoview platform or extracting data from it.
Neoview Workload Management Services Guide	Information about using Neoview Workload Management Services (WMS) to manage workload and resources on a Neoview data warehousing platform.
README files for Administration products	— README for the HP Neoview Management Dashboard Client — README for HP Neoview Command Interface — README for HP Neoview Reports Client — README for the Neoview Performance Analyzer

- **Reference**

Neoview SQL Reference Manual	Reference information about the syntax of SQL statements, functions, and other SQL language elements supported by the Neoview database software.
Mapping Tables for Neoview Character Sets	Provides links to the mapping tables used by the Neoview Character Sets product.
Neoview Messages Manual	Cause, effect, and recovery information for error messages.

- **Connectivity**

Neoview JDBC Type 4 Driver API Reference	Reference information about the HP Neoview JDBC Type 4 Driver API.
Neoview JDBC Type 4 Driver Programmer's Reference	Information about using the HP Neoview JDBC Type 4 driver, which provides Java applications on client workstations access to a Neoview database.
Neoview ODBC Drivers Manual	Information about using HP Neoview ODBC drivers on a client workstation to access a Neoview database.
ODBC Client Administrator Online Help	Context-sensitive help topics that describe how to use the ODBC client interface.
README files for Connectivity products	— README for the HP Neoview JDBC Type 4 Driver — README for the HP Neoview ODBC Driver for Windows — README for the HP Neoview UNIX Drivers

Publishing History

Part Number	Product Version	Publication Date
544357-002	HP Neoview Release 2.0 *(beta)*	April 2007
544587–001	HP Neoview Release 2.2 *(beta)*	August 2007
544806–001	HP Neoview Release 2.3	April 2008

HP Encourages Your Comments

HP encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

pubs.comments@hp.com

Include the document title, part number, and any comment, error found, or suggestion for improvement you have concerning this document.

1 Introducing Neoview Workload Management Services (WMS)

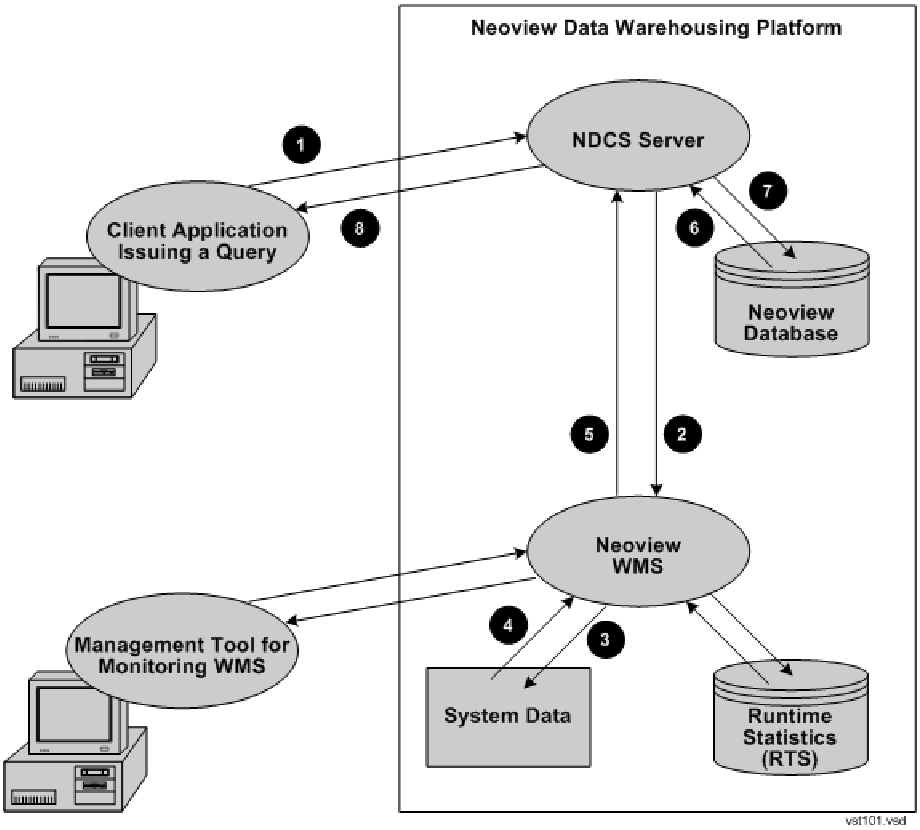
The Neoview Workload Management Services (WMS) feature provides the infrastructure to help you manage system and service-level resources for executing queries on a Neoview platform. You can configure service levels in WMS to prioritize and set rules for groups of queries (that is, query workloads) executing on the Neoview platform. You can then monitor queries in service levels, identify problematic queries, hold queries to free resources, and cancel runaway queries to prevent them from monopolizing system resources.

WMS supports management tools, such as the Neoview Command Interface (NCI) for executing WMS commands and the Neoview Management Dashboard Client for monitoring active queries in a graphical user interface. Those management tools enable you to monitor and manage queries in WMS from a client workstation.

How WMS Works

WMS acts as a query manager for Neoview Database Connectivity Service (NDCS) server instances, obtaining information from the requesting NDCS servers and using that information to manage query workload. WMS monitors queries that are submitted to NDCS services from various client applications, such as JDBC or ODBC client applications or the NCI command-line interface. WMS manages both prepared queries and queries that are executed directly. However, WMS does not manage singleton queries, which access and return only one row of data. Singleton queries run directly against the database without being managed by WMS. If a query accesses multiple rows, even if the query returns only one row of data, such as in the case of SELECT COUNT(*), WMS manages that query. Figure 1-1 shows how a query is executed through WMS. The numbers in the diagram show the sequence in which query requests are handled by WMS.

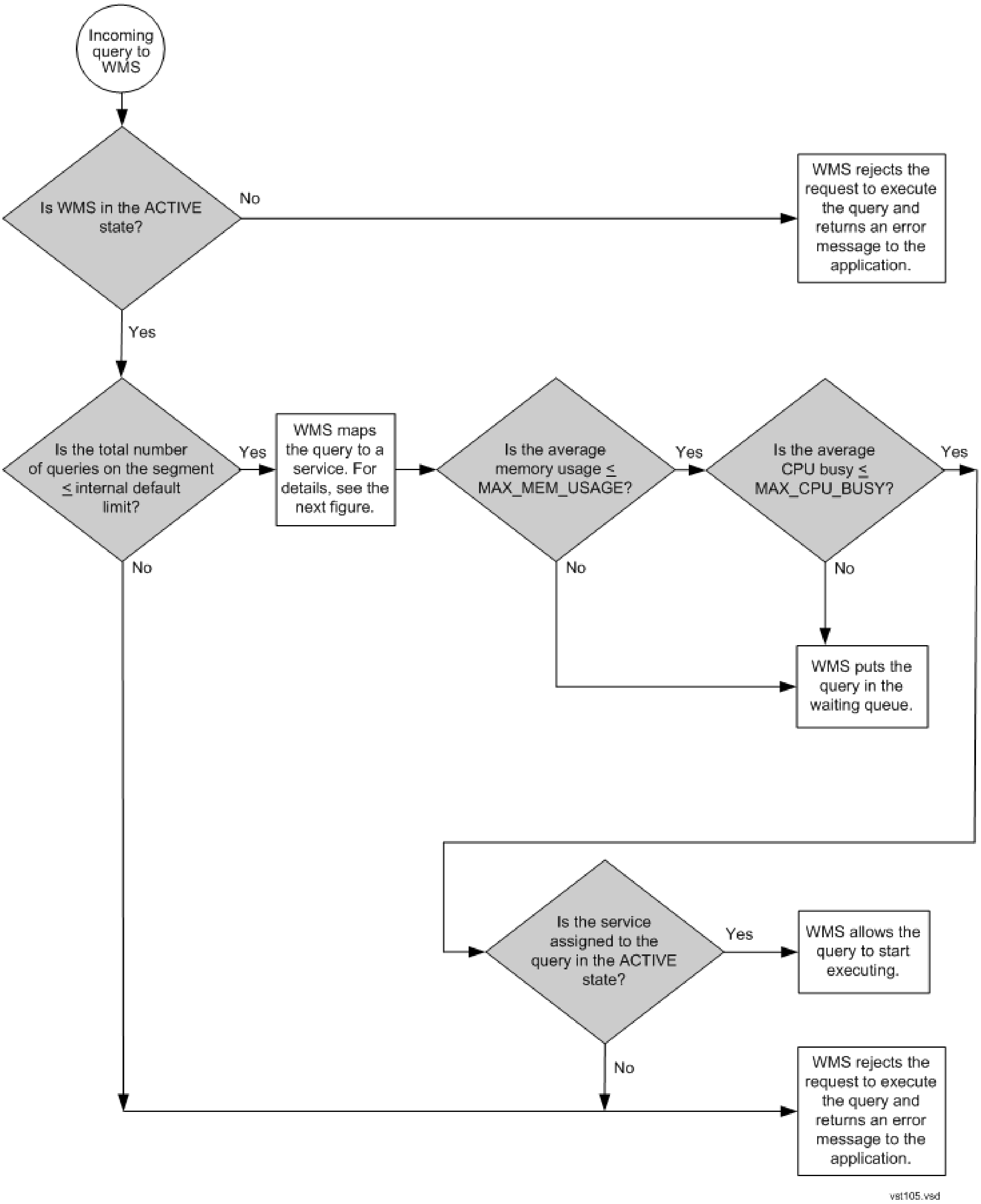
Figure 1-1 Execution of Queries Through WMS



Neoview Data Warehousing Platform
NDCS Server
1
8
Client Application Issuing a Query
7
6
Neoview Database
5
2
Neoview WMS
4
3
Management Tool for Monitoring WMS
System Data
Runtime Statistics (RTS)
vst101.vsd

When a query is submitted to the Neoview platform, the NDCS server passes the execution decision making to WMS. Figure 1-2 shows how WMS handles an incoming query.

Figure 1-2 How WMS Handles an Incoming Query

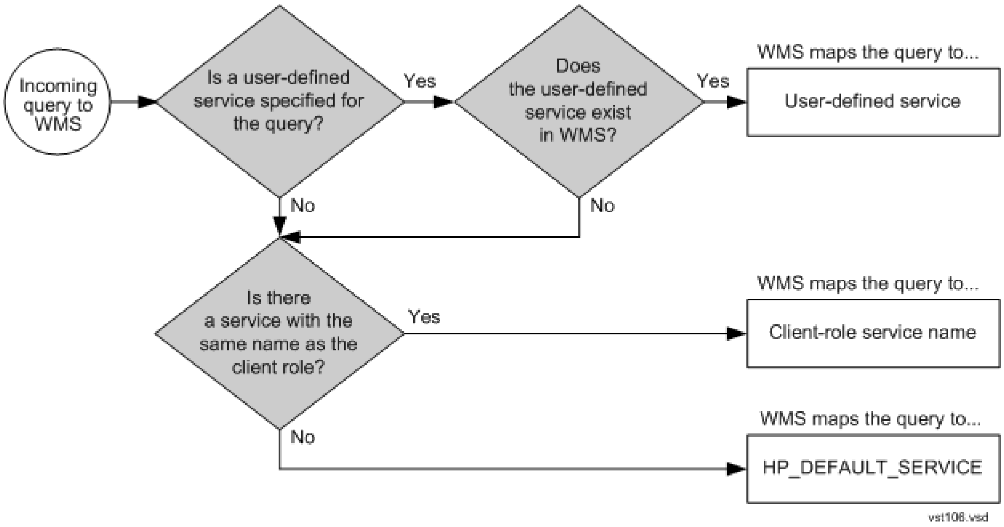

Incoming query to WMS
Is WMS in the ACTIVE state?
No
WMS rejects the request to execute the query and returns an error message to the application.
Yes
Is the total number of queries on the segment ≤ internal default limit?
Yes
WMS maps the query to a service. For details, see the next figure.
Is the average memory usage ≤ MAX_MEM_USAGE?
Yes
Is the average CPU busy ≤ MAX_CPU_BUSY?
Yes
No
No
No
WMS puts the query in the waiting queue.
Is the service assigned to the query in the ACTIVE state?
Yes
WMS allows the query to start executing.
No
WMS rejects the request to execute the query and returns an error message to the application.
vst105.vsd

When handling an incoming query, WMS first checks that WMS is in the active state. WMS is inactive when a user places it on hold by using the HOLD command. For more information, see the "HOLD Command" (page 63). If WMS is inactive (that is, on hold), WMS rejects the request to execute the query and returns an error message to the application. If WMS is active, WMS checks that the total number of executing and waiting queries in WMS is less than or equal to the internal default limit of 1000 per segment. If the total number of queries in WMS exceeds the limit, WMS rejects the request to execute the query and returns an error message to the application.

If the total number of queries in WMS is less than or equal to the limit, WMS maps the query to a service level.

WMS maps the query to a service level based on the attribute set by the NDCS server. If you set a service for the client session or the query, WMS executes the query in that service. If you do not set a service for the client session or query, WMS searches for a service name that matches the second part of the client role, such as MGR from ROLE.MGR. If WMS does not find a service name that matches the role, WMS maps the query to the default service, HP_DEFAULT_SERVICE. See Figure 1-3 (page 19).

Figure 1-3 How WMS Maps a Service to a Query



After WMS maps the query to a service level, it checks that the average memory usage is less than or equal to the threshold limit, MAX_MEM_USAGE, of the service. If the MAX_MEM_USAGE was not set for the service, WMS checks the MAX_MEM_USAGE of the WMS system configuration. If the MAX_MEM_USAGE was not set for the WMS system configuration, WMS checks the internal default threshold for MAX_MEM_USAGE, which is 85% . If the average memory usage exceeds the MAX_MEM_USAGE, WMS puts the query in the waiting queue.

If the average memory usage is less than or equal to the MAX_MEM_USAGE, WMS checks that the average CPU busy is less than or equal to the threshold limit, MAX_CPU_BUSY, of the service. If the MAX_CPU_BUSY was not set for the service, WMS checks the MAX_CPU_BUSY of the WMS system configuration. If the MAX_CPU_BUSY was not set for the WMS system configuration , WMS checks the internal default threshold for MAX_CPU_BUSY, which is 100%. If the average CPU busy exceeds the MAX_CPU_BUSY, WMS puts the query in the waiting queue.

WMS continues to check the queries in the waiting queue to see if they can execute given the current system resources (memory usage and CPU busy). As soon as resources become available, WMS starts executing the queries associated with higher priority services in the waiting queue. For information about how the priority of a service influences the execution of queries in that service, see "How Service Priorities and System Resource Thresholds Work" (page 25).

If the average memory usage and CPU busy are under the resource threshold limits, WMS checks that the service assigned to the query is in the active state. The active state is influenced by the active time of the service or by the HOLD command. For example, if the active time of a service is between the hours of 22:00 and 23:59, the service is inactive (that is, on hold) any time outside of 22:00 and 23:59. Even if the current time is within the active time of the service, a user can put the service on hold by issuing the HOLD command. For more information, see the "HOLD Command" (page 63). If the service assigned to the query is inactive (that is, on hold), WMS rejects the request to execute the query and returns an error message to the application. If the service is active, WMS allows the query to start executing.

WMS obtains compile-time information about the query from the NDCS server, such as the query ID, CPU time, I/O time, messages time, idle time, total time, and cardinality. At preconfigured intervals, WMS also captures system resource information, such as the CPU busy, memory usage, query cache hits, and disk I/O, and also captures query runtime statistics.

How to Use WMS

Using WMS commands, you can set properties at the system level or at the service level to manage resource consumption and query workloads. At the system level, WMS monitors queries based on CPU and memory usage. At the service level, WMS monitors queries based on the thresholds that you set for the service, such as the priority, CPU busy, memory usage, and active time. For more information, see:

- "Setting Up Services and Thresholds in WMS" (page 20)
- "Setting System Thresholds in WMS" (page 20)
- "Monitoring and Managing Queries in WMS" (page 20)

Setting Up Services and Thresholds in WMS

Using WMS commands, you can configure service levels in WMS for queries that run on the Neoview platform. Each service level has a priority and a set of rules, or threshold limits, which specify the maximum CPU busy, the maximum memory usage, and the active period of query execution for the service.

WMS manages queries associated with a service level according to the priority and thresholds of the service level. If a service level has a low priority, the queries associated with the low-priority service level are executed later in the queue. If a service level has an active period from 20:00 to 23:00, WMS prevents queries associated with the service from starting to execute outside the time range by rejecting those queries. If the maximum CPU busy or memory usage is reached for the service, WMS puts new queries in the waiting queue.

To configure WMS with service levels and thresholds, see Chapter 3 (page 25).

Setting System Thresholds in WMS

You can also configure system thresholds in WMS, such as the maximum CPU busy, the maximum memory usage, and the interval for collecting query runtime statistics. For more information, see "Configuring the WMS System Configuration" (page 33).

If the CPU or memory usage exceeds the maximum limit, WMS prevents waiting queries from being executed to prevent additional CPU or memory from being used and to prevent those queries from affecting the currently executing queries. New queries are put in the waiting queue.

Monitoring and Managing Queries in WMS

To manage query workloads according to the service levels that you configured in WMS, you must associate the queries with a service name. There are specific methods and functions that enable you to associate queries with a service level in JDBC and ODBC client applications. If you are executing queries in NCI, use the SET SERVICE statement to set the service level of the NCI session. For more information about associating queries with services, see Chapter 4 (page 35).

You can monitor the queries and services being managed by WMS by using WMS commands in client applications, such as NCI. The WMS command, STATUS, returns information to the client application about the queries and services being managed by WMS. The WMS commands, HOLD, RELEASE, and CANCEL, enable you to hold queries that are in the waiting queue, release holding queries, and cancel queries to free resources, respectively. For more information about using WMS commands to monitor and manage query workloads, see Chapter 5 (page 41).

You can also monitor the queries being managed by WMS by using the Neoview Management Dashboard Client. The QueryRTS entity in the Dashboard Client displays the queries being managed by WMS in a graphical user interface. The Command and Control feature of the

Dashboard Client enables you to suspend (hold) , resume (release), and kill (cancel) queries displayed in the QueryRTS entity. For more information, see the *Neoview Management Dashboard Client Guide for Database Administrators*.

2 Getting Started

Installing and Starting WMS on the Neoview Platform

If you have Neoview Release 2.3, WMS should already be installed and started on your Neoview platform. If you are an HP support person, see the *Neoview Database Support Guide* for information about installing and starting WMS on a Neoview platform.

Installing the Neoview Command Interface (NCI)

If you plan to execute the WMS commands in a command-line interface or in script files off the Neoview platform, install the NCI product on your client workstation. For installation instructions, see the *Neoview Command Interface (NCI) Guide*.

3 Configuring WMS

How Services Work

What Are Services?

Service levels are a mechanism in WMS to enable you to prioritize query workloads on a Neoview platform and to partition available workload resources. Each service level in WMS has a priority and system resource thresholds associated with it. By default, WMS uses the default service, HP_DEFAULT_SERVICE, to handle incoming queries. You can alter the thresholds of HP_DEFAULT_SERVICE but cannot delete it from the system.

You can define your own services to partition query workload resources according to your workload management strategy. You may add up to 16 user-defined services in addition to HP_DEFAULT_SERVICE on a Neoview platform. User-defined services have these attributes:

PRIORITY	The priority of the service relative to other services. The default is MEDIUM.
MAX_CPU_BUSY	A percentage from zero to 100 of the total CPU busy allowed for the service. This value is an average of CPU busy across all segments in the cluster. The default is zero, meaning that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE	A percentage from zero to 100 of memory usage allowed for the service. This value is an average of memory usage across all segments in the cluster. The default is zero, meaning that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.
ACTIVE	The time period during which the service is active and can execute queries. The default is 00:00 through 23:59.
EXEC_TIMEOUT	The number of minutes a query is allowed to remain in the executing state before WMS cancels the query. The default is zero, meaning no timeout.
WAIT_TIMEOUT	The number of minutes a query is allowed to remain in the waiting state before WMS cancels the query. The default is zero, meaning no timeout.
PLAN or NO_PLAN	Either collects the query execution plans or not so that client tools can use and display that data. The default is NO_PLAN.
TEXT or NO_TEXT	Either collects the query syntax or not so that client tools can use and display that data. The default is NO_TEXT.
COMMENT	Text that describes the service. By default, there is no comment.

To change the attributes of a user-defined service, use the ALTER SERVICE command. For more information, see “Altering a Service” (page 31) and the “ALTER SERVICE Command” (page 58).

How Service Priorities and System Resource Thresholds Work

WMS controls the execution of queries based on the service priority. The priority of a service determines when the queries associated with the service are executed in relation to queries associated with other services. If a service has a lower priority and if the queries associated with higher priority services are currently executing, the queries associated with the lower priority service execute next, provided that the system resource usage has not reached its limits.

WMS throttles incoming queries (that is, puts them into a waiting queue) when the system resource usage reaches its limits. To determine available CPU and memory resources for the system, WMS looks at a moving average every five seconds for six intervals of CPU busy and takes a snapshot every five seconds of memory usage on the system. By default, the system resource thresholds are a maximum CPU busy of 100% and a maximum memory usage of 85%. However, you can change those system resource thresholds either at the WMS level by altering WMS or at the service level by adding or altering a user-defined service. A user-defined service's thresholds override the thresholds of the WMS system configuration. The thresholds of the WMS system configuration override the internal default thresholds.

Service Priority Weights

WMS assigns weights to each service priority:

Service Priority	Weight
URGENT	6
HIGH	5
MEDIUM-HIGH	4
MEDIUM	3
LOW-MEDIUM	2
LOW	1

Calculation of Expected Query Execution Percentage

WMS determines an expected percentage of executing queries for each service based on the service's priority weight:

```
Expected-percentage = (priority-weight / total-weight) * 100
```

The `priority-weight` is the weight of the service. For example, a service with a HIGH priority has a `priority-weight` of 5. The `total-weight` is the sum of the weights of all active services. For example, if there are three active services that have HIGH, MEDIUM, and LOW priorities, the `total-weight` is the sum of `5 + 3 + 1`, which is 9. The expected percentage for a service with a HIGH priority is `5/9 * 100`, which is 56%.

Example 1: How Service Priorities and System Resource Thresholds Work

Suppose that you have these service levels:

Service Name	Priority	Expected Percentage Based on Priority Weight
SERVICE1	HIGH	5/9 * 100 = 56%
SERVICE2	MEDIUM	3/9 * 100 = 33%
SERVICE3	LOW	1/9 * 100 = 11%

Consider the system resource usage when queries associated with those services request to be executed on the system:

Service Name	Total Queries	CPU Busy	Maximum CPU Busy	Memory Usage	Maximum Memory Usage
SERVICE1	80	50%	100%	60%	85%
SERVICE2	20	50%	100%	60%	85%
SERVICE3	25	50%	100%	60%	85%

Because the system resources are within the limits (100% for CPU busy and 85% for memory usage), WMS allows all incoming queries to start executing. If ten additional queries request to be executed in the low-priority service, SERVICE3, and the system resources reach their limits when the fifth query starts executing, WMS puts the five remaining queries into the waiting queue:

Service Name	Total Queries	Executing Queries	Waiting Queries	CPU Busy (Limit = 100%)	Memory Usage (Limit = 85%)
SERVICE1	80	80	0	90%	86%
SERVICE2	20	20	0	90%	86%
SERVICE3	25+10=35	25+5=30	0+5=5	90%	86%

Of the incoming queries, if 20 queries associated with the high-priority SERVICE1, 10 queries associated with the medium-priority SERVICE2, and five queries associated with the low-priority SERVICE3 request to be executed, WMS puts all those queries into the waiting queue:

Service Name	Total Queries	Executing Queries	Waiting Queries	CPU Busy (Limit = 100%)	Memory Usage (Limit = 85%)
SERVICE1	80+20=100	80	0+20=20	90%	86%
SERVICE2	20+10=30	20	0+10=10	90%	86%
SERVICE3	35+5=40	30	5+5=10	90%	86%

As queries in each of the services finish executing, system resources free up, and WMS dynamically balances the waiting and executing queries based on the percentage of executing queries for each service and how the percentage compares with the expected percentage, which is based on the service's priority weight. For example, suppose that 15 queries finish executing in SERVICE1:

Service Name	Total Queries	Executing Queries	Percentage of All Executing Queries	Expected Percentage Based on Priority Weight	Waiting Queries
SERVICE1	100–15=85	80–15=65	65/115 = 56%	56%	20
SERVICE2	30	20	20/115 = 17%	33%	10
SERVICE3	40	30	30/115 = 26%	11%	10

For each service, WMS calculates the percentage of all executing queries and compares it with the expected percentage for the service. If the percentage of executing queries is equal to or lower than the expected percentage, WMS allows the queries in the waiting queue to start executing, starting with queries in a higher priority service and ending with queries in a lower priority service. For example, SERVICE1 queries have a higher priority than SERVICE3 queries and start executing first. In this example, when 15 queries in SERVICE1 finish executing, WMS allows five queries in SERVICE1 and then 10 queries in SERVICE2 to start executing:

Service Name	Total Queries	Executing Queries	Percentage of All Executing Queries	Expected Percentage Based on Priority Weight	Waiting Queries
SERVICE1	85	65+5=70	70/130 = 54%	56%	20–5=15
SERVICE2	30	20+10=30	30/130 = 23%	33%	10–10=0
SERVICE3	40	30	30/130 = 23%	11%	10

Example 2: How Service Priorities and System Resource Thresholds Work

Consider another example where there are 30 incoming queries associated with the low-priority SERVICE3. The system resources reach their limits when the twentieth query of SERVICE3 starts executing. WMS puts the ten remaining queries into the waiting queue.

Service Name	Total Queries	Executing Queries	Waiting Queries	CPU Busy (Limit = 100%)	Memory Usage (limit = 85%)
SERVICE1	0	0	0	90%	86%
SERVICE2	0	0	0	90%	86%
SERVICE3	30	20	10	90%	86%

When the system resource limits are met, WMS puts additional incoming queries into the waiting queue. For example, if 60 queries associated with the high-priority SERVICE1 and 30 queries associated with the medium-priority SERVICE2 come in to the system, WMS puts all those queries into the waiting queue:

Service Name	Total Queries	Executing Queries	Waiting Queries	CPU Busy (Limit = 100%)	Memory Usage (limit = 85%)
SERVICE1	0+60=60	0	0+60=60	90%	86%
SERVICE2	0+30=30	0	0+30=30	90%	86%
SERVICE3	30	20	10	90%	86%

As queries finish executing, WMS dynamically balances the waiting and executing queries and moves some queries from the waiting queue to the executing state based on the expected percentage of executing queries and the priority of the service. For example, suppose that 10 queries finish executing in SERVICE3:

Service Name	Total Queries	Executing Queries	Percentage of All Executing Queries	Expected Percentage Based on Priority Weight	Waiting Queries
SERVICE1	60	0	0/10 = 0%	56%	60
SERVICE2	30	0	0/10 = 0%	33%	30
SERVICE3	30–10=20	20–10=10	10/10 = 100%	11%	10

If the percentage of executing queries in a service is equal to or lower than the expected percentage, WMS allows the queries in the waiting queue to start executing, starting with queries in the high-priority SERVICE1 and ending with the queries in a lower priority service. For example, when 10 low-priority SERVICE3 queries finish executing, WMS allows 50 high-priority SERVICE1 queries and then 30 medium-priority SERVICE2 queries in the waiting queue to start executing until the system resources reach their limits:

Service Name	Total Queries	Executing Queries	Percentage of All Executing Queries	Expected Percentage Based on Priority Weight	Waiting Queries
SERVICE1	60	0+50=50	50/90 = 56%	56%	60–50=10
SERVICE2	30	0+30=30	30/90 = 33%	33%	30–30=0
SERVICE3	30	10	10/90 = 11%	11%	10

Effect of Service-Level Thresholds

For each service level, you can set system resource thresholds, which take precedence over the thresholds of the WMS system configuration. For example, for a lower priority service, such as SERVICE3, suppose you set the memory usage to 20% and the CPU busy to 50%. Incoming queries associated with SERVICE3 automatically go into the waiting queue when the maximum memory usage and CPU busy exceed the service thresholds of 20% and 50%, respectively. For more information, see "Configuring User-Defined Services" (page 29).

Setting lower system resource thresholds for lower priority services restricts the number of executing queries associated with lower priority services and enables more queries associated with higher priority services to execute on the Neoview platform. This strategy might be useful for preventing lower priority queries, such as from reporting or ad-hoc applications, from consuming the system resources needed by higher priority queries.

Configuring User-Defined Services

This section describes how to establish user-defined services for managing query workloads in WMS. For more information about services, see "How Services Work" (page 25). The examples show WMS commands that are executed in an NCI session. For more information, see "How to Execute WMS Commands in NCI" (page 55).

To add, alter, delete, or display information about a service, see:

- "Adding a Service" (page 29)
- "Altering a Service" (page 31)
- "Deleting a Service" (page 32)
- "Displaying Information About the Service Definitions" (page 32)

Adding a Service

Use the ADD SERVICE command to add a service and its priority level and thresholds to the WMS configuration on a Neoview platform.

Naming the Service

Select a unique and meaningful name for the service. You cannot add a service that already exists, such as SYSTEM or HP_DEFAULT_SERVICE. You can specify a service name of up to 24 characters. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

Setting the Priority of the Service

You can set the priority of the service relative to other services. Choose one of these priorities, from highest to lowest:

- PRIORITY URGENT
- PRIORITY HIGH

- PRIORITY MEDIUM-HIGH
- PRIORITY MEDIUM
- PRIORITY LOW-MEDIUM
- PRIORITY LOW

If you do not specify a priority for the service, the default priority is medium.

Setting the System Resource Thresholds of the Service

You can set these system resource thresholds for the service:

- MAX_CPU_BUSY, which is a percentage from zero to 100 of the total CPU busy allowed for the service. This value is an average of CPU busy across all segments in the cluster.
- MAX_MEM_USAGE, which is a percentage from zero to 100 of memory usage allowed for the service. This value is an average of memory usage across all segments in the cluster.

If you do not set system resource thresholds for the service, WMS uses the system resource thresholds of the WMS system configuration. If you do not set the system resource thresholds of the WMS system configuration, WMS uses these internal default thresholds:

- 100% for MAX_CPU_BUSY
- 85% for MAX_MEM_USAGE

Setting the Active Time of the Service

The active time of a service is a time range when the service can execute queries. Outside the active time range, a service cannot execute incoming queries associated with the service. If a query is still executing in the service when the end time is reached, WMS allows the query to continue executing outside the active time range. However, WMS rejects any new incoming queries associated with the service if they occur outside the active time range.

By default, the active time is from 00:00 through 23:59. To change the active time, specify the start and end times during which you want the service to execute queries. The start or end time, *hh*:*mm*, is in hours (*hh*) and minutes (*mm*). Currently, the active time cannot span more than one day.

For example, if you specify an active time of 15:00 to 18:00, WMS enables queries associated with the service to start executing between the hours of 15:00 to 18:00. WMS rejects any queries associated with the service that try to execute outside the active time of 15:00 to 18:00.

Setting the Execution and Waiting Timeouts of the Service

You can set a timeout value, EXEC_TIMEOUT, for executing queries in the service. The timeout value can be from zero to 1440 minutes. When the timeout value is reached, WMS cancels the executing query and stops that NDCS server. If you do not specify an EXEC_TIMEOUT value, the default is zero, meaning no timeout is in effect.

△ **CAUTION:** Do not set an EXEC_TIMEOUT value for a service that handles long-running inserts or updates of the database.

You can also set a timeout value, WAIT_TIMEOUT, for the waiting queries in the service. The timeout value can be from zero to 1440 minutes. When the timeout value is reached, WMS cancels the waiting query and returns an error message. If you do not specify a WAIT_TIMEOUT value, the default is zero, meaning no timeout is in effect.

Setting the Output Options of the Service

Specify the PLAN option to collect the execution plan of queries executing in the service. Client tools, such as NCI and the Neoview Manageability Repository, can then use and display this data. If you do not specify PLAN, WMS does not collect any plans for the service (that is, NO_PLAN) .

Specify the TEXT option to collect the SQL syntax of queries executing in the service. Client tools, such as NCI and the Neoview Manageability Repository can then use and display this data. If you do not specify TEXT, WMS does not collect any SQL text for the service (that is NO_TEXT).

Adding a Comment for the Service

You can optionally provide text to describe the service that you are creating. For example, to describe a service intended for data loading, you might specify:

```
"dataloading service"
```

You must delimit the comment in double quotes. For details about the syntax, see "ADD SERVICE Command" (page 56).

Example of Adding a Service

For example, in NCI, this ADD SERVICE command adds the DATALOADING service to the WMS configuration, gives the service high priority, and sets the active period of queries to 20:00 to 23:00:

```
NS%add service dataloading priority high, active 20:00 to 23:00;

--- WMS operation complete.

NS%
```

For the syntax, see the "ADD SERVICE Command" (page 56). You can also execute the ADD SERVICE command in JDBC and ODBC client applications. For more information, see "How to Execute WMS Commands in NCI" (page 55).

Altering a Service

Use the ALTER SERVICE command to alter the setting of a service. For example, in NCI, this ALTER SERVICE command changes the priority and active time of the DATALOADING service in the WMS configuration:

```
NS%alter service dataloading priority medium-high, active 22:00 to 23:59;

--- WMS operation complete.

NS%
```

For the syntax, see the "ALTER SERVICE Command" (page 58). You can also execute the ALTER SERVICE command in JDBC and ODBC client applications. For more information, see "How to Execute WMS Commands in NCI" (page 55).

How an Altered Service Affects Queries

Whenever you alter a service, it is recommended that you do so when few or no queries associated with that service are running on the Neoview platform. In case you alter a service when queries associated with that service are being issued, be aware of these consequences.

New values of the maximum CPU busy and maximum memory usage do not affect the currently executing queries associated with the service. However, new values of the maximum CPU busy and maximum memory usage affect waiting queries and new incoming queries associated with the service. When the new maximum CPU busy and memory usage are less than the previous values of the service, the lower thresholds cause waiting queries associated with the service to stay in the waiting queue longer until resources become available. When the new maximum CPU busy and memory usage are greater than the previous values of the service, the higher thresholds cause waiting queries associated with the service to start executing sooner than they would have at lower thresholds.

Changing the priority or active time of the service does not affect the currently executing queries associated with the service. However, the new priority or active time does affect waiting queries and any new incoming queries associated with the service.

Changing the EXEC_TIMEOUT, WAIT_TIMEOUT, PLAN, and TEXT parameters does not affect the currently executing or waiting queries associated with the service. However, changing these settings does affect new incoming queries associated with the service.

Deleting a Service

Use the DELETE SERVICE command to delete a service from the WMS configuration. After you delete a service, incoming queries associated with the deleted service are managed either by a service with the same name as the client role, if one exists, or by the HP_DEFAULT_SERVICE service if a service with the same name as the client role does not exist. For example, in NCI, this DELETE SERVICE command removes the DATALOADING service from the WMS configuration:

```
NS%delete service dataloading;

--- WMS operation complete.

NS%
```

For the syntax, see the "DELETE SERVICE Command" (page 62). You can also execute the DELETE SERVICE command in JDBC and ODBC client applications. For more information, see "How to Execute WMS Commands in NCI" (page 55).

When you issue a DELETE SERVICE command, the state of the deleted service changes from active or hold to deleting. In the deleting state, WMS cancels all waiting queries and rejects all incoming queries (that is, execution requests) associated with the service. To cancel currently executing queries associated with the deleted service, use the IMMEDIATE option with the DELETE SERVICE command. The IMMEDIATE option cancels all currently executing queries associated with the service when you delete the service. If you do not specify the IMMEDIATE option, WMS allows the queries to finish executing.

Displaying Information About the Service Definitions

The INFO SERVICE command displays information about the configuration of one or all services in WMS:

SERVICE_NAME	Name of the service
SERVICE_PRIORITY	The priority of the service relative to other services. The default is MEDIUM.
MAX_CPU_BUSY	A percentage from zero to 100 of the total CPU busy allowed for the service. This value is an average of CPU busy across all segments in the cluster. The default is zero, meaning that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE	A percentage from zero to 100 of memory usage allowed for the service. This value is an average of memory usage across all segments in the cluster. The default is zero, meaning that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.
ACTIVE_TIME	The time period during which the service is active and can execute queries. The default is 00:00 through 23:59.
SQL_PLAN	Displays PLAN to indicate that the query execution plans are collected for display or NO_PLAN to indicate that no plans are collected for display
SQL_TEXT	Displays TEXT to indicate that the SQL syntax of executing queries is collected for display or NO_TEXT to indicate that no SQL text is collected for display
EXEC_TIMEOUT	The number of minutes a query is allowed to remain in the executing state before WMS cancels the query. The default is zero, meaning no timeout.
WAIT_TIMEOUT	The number of minutes a query is allowed to remain in the waiting state before WMS cancels the query. The default is zero, meaning no timeout.
COMMENT	Text that describes the service. By default, there is no comment.

For example, in NCI, this INFO SERVICE command displays configuration information for all services in WMS:

```
NS%info service all;

SERVICE_NAME                                       SERVICE_PRIORITY MAX_CPU_BUSY MAX_MEM_USAGE ACTIVE_TIME
                                    SQL_PLAN SQL_TEXT EXEC_TIMEOUT WAIT_TIMEOUT COMMENT


-------------------------------------------------- ---------------- ------------ -------------
-------------------------------------------------- -------- -------- ------------ ------------
--------------------------------------------------------------------------------------------------------------
HP_DEFAULT_SERVICE                                 LOW                         0             0 FROM 00:00 TO
23:59                                NO_PLAN  NO_TEXT             0            0 DEFAULT SERVICE


SERVICE1                                           MEDIUM                      0             0 FROM 00:00 TO
23:59                                NO_PLAN  NO_TEXT             0            0


--- WMS operation complete.

NS%
```

This INFO SERVICE command in NCI displays configuration information for the HP_DEFAULT_SERVICE service:

```
NS%info service hp_default_service;

SERVICE_NAME                                       SERVICE_PRIORITY MAX_CPU_BUSY MAX_MEM_USAGE ACTIVE_TIME
                                    SQL_PLAN SQL_TEXT EXEC_TIMEOUT WAIT_TIMEOUT COMMENT


-------------------------------------------------- ---------------- ------------ -------------
-------------------------------------------------- -------- -------- ------------ ------------
--------------------------------------------------------------------------------------------------------------
HP_DEFAULT_SERVICE                                 LOW                         0             0 FROM 00:00 TO
23:59                                NO_PLAN  NO_TEXT             0            0 DEFAULT SERVICE


--- WMS operation complete.

NS%
```

For the syntax, see the “INFO SERVICE Command” (page 64). You can also execute the INFO SERVICE command in JDBC and ODBC client applications. For more information, see “How to Execute WMS Commands in NCI” (page 55).

Configuring the WMS System Configuration

WMS has a built-in system configuration that defines the maximum CPU busy, maximum memory usage, and statistics collection interval for the cluster. As a database administrator, you can alter these attributes of the WMS system configuration:

MAX_CPU_BUSY	The percentage from zero to 100 of the total CPU busy allowed for the system. This value is an average of CPU busy across all segments in the cluster. If you do not set this threshold, the default is zero, meaning that WMS uses the internal default value of 100%.
MAX_MEM_USAGE	The percentage from zero to 100 of memory usage allowed for the system. This value is an average of memory usage across all segments in the cluster. If you do not set this threshold, the default is zero, meaning that WMS uses the internal default value of 85% .
STATS_INTERVAL	An interval in seconds at which WMS collects runtime statistics for all executing queries managed by WMS. The default is five seconds.

To change the attributes of the WMS system configuration, use the ALTER WMS command. See .

Altering the WMS System Configuration

Use the ALTER WMS command to alter the WMS system configuration (that is, the SYSTEM service), which includes the maximum CPU busy, the maximum memory usage, and the refresh interval for query runtime statistics.

- For MAX_CPU_BUSY, specify a percentage for the maximum CPU busy of all CPUs on the system. The default value is zero, meaning that WMS uses the internal value of 100 percent.
- For MAX_MEM_USAGE, specify a percentage for the maximum memory usage on the system. The default value is zero, meaning that WMS uses the internal value of 85 percent.
- For STATS_INTERVAL, specify an interval in seconds at which WMS collects runtime statistics for all executing queries managed by WMS. The value must be greater than or equal to five seconds and less than or equal to 300 seconds. The default value is five seconds.

For example, in NCI, this ALTER WMS command changes the MAX_CPU_BUSY and MAX_MEM_USAGE of the WMS system configuration:

```
NS%alter wms max_cpu_busy 80, max_mem_usage 90;

--- WMS operation complete.

NS%
```

For the syntax, see the “ALTER WMS Command” (page 60). You can also execute the ALTER WMS command in JDBC and ODBC client applications. For more information, see “How to Execute WMS Commands in NCI” (page 55).

Displaying the WMS System Configuration

The INFO WMS command displays information about the WMS system configuration:

MAX_CPU_BUSY	A percentage representing the maximum CPU busy of all CPUs on the system. If INFO WMS displays zero in the output, it means that WMS uses the internal value of 100 percent.
MAX_MEM_USAGE	A percentage representing the maximum memory usage on the system. If INFO WMS displays zero in the output, it means that WMS uses the internal value of 85 percent.
STATS_INTERVAL	An interval in seconds at which WMS collects runtime statistics for all executing queries managed by WMS. The default value is five seconds.

For example, in NCI, this INFO WMS command displays information about the WMS system configuration:

```
NS%info wms;

MAX_CPU_BUSY MAX_MEM_USAGE STATS_INTERVAL
------------ ------------- --------------
           0             0              5

--- WMS operation complete.

NS%
```

The zeros for the MAX_CPU_BUSY and MAX_MEM_USAGE mean that WMS is using the internal default values of 100% and 85%, respectively.

For the syntax, see the“INFO WMS Command” (page 65). You can also execute the INFO WMS command in JDBC and ODBC client applications. For more information, see “How to Execute WMS Commands in NCI” (page 55).

4 Associating Queries With Services

If you do not associate an SQL query with a service, WMS first looks for a service name that matches the second part of the client role, such as MGR in ROLE.MGR. If WMS finds a service name that matches the client role, WMS manages the query in that service. If WMS does not find a service name that matches the client role, WMS manages the query in the default service, HP_DEFAULT_SERVICE.

If you want to manage query workloads using specific services, you must explicitly associate the queries with those services. This chapter explains how to associate queries with services by using these approaches, from lowest to highest precedence:

- “Specifying a Service in a Client Data Source” (page 35)
- “Using the SQLSetConnectAttr() Function to Associate Queries With a Service” (page 37)
- “Using the setServiceName() Method to Associate Queries With a Service” (page 37)
- “Using the SET SERVICE Statement to Associate Queries With a Service” (page 38)

NOTE: The service name is not case-sensitive and can contain multibyte characters. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

Specifying a Service in a Client Data Source

You can specify a service for a client data source by using one these techniques:

- “Setting the Service in the ODBC Administrator on a Windows Workstation” (page 35)
- “Setting the Service in the MXODSN File on a Linux or UNIX Workstation” (page 37)

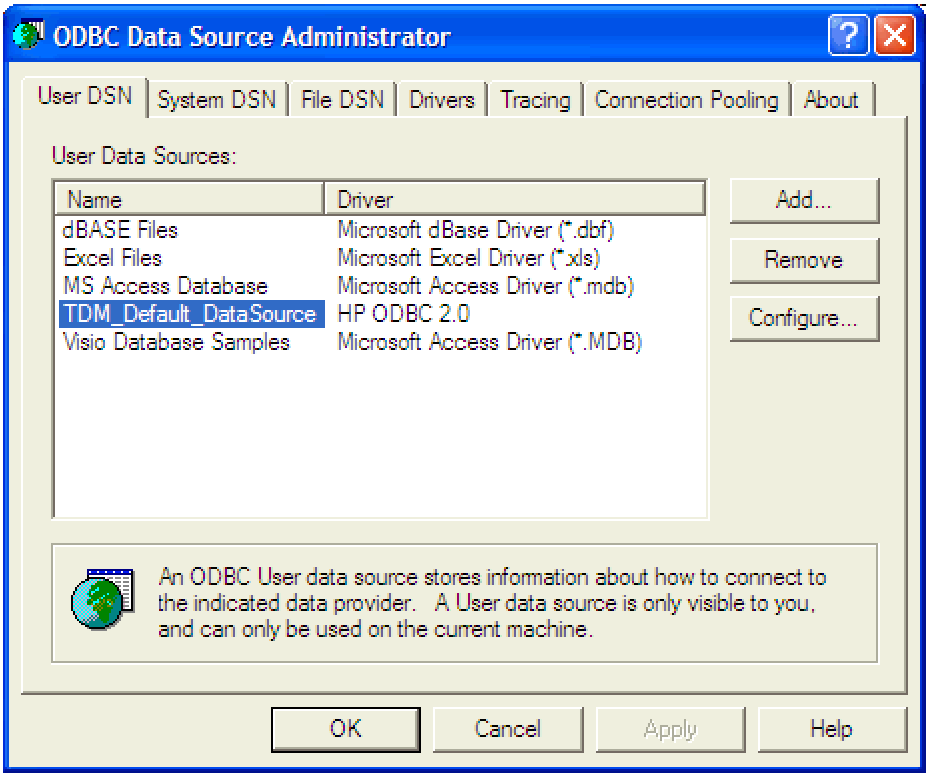
Setting the Service in the ODBC Administrator on a Windows Workstation

On a Windows workstation, you can set the service name for a client data source by using the ODBC Data Source Administrator. For more information, see the ODBC Client Administrator Online Help.

NOTE: The ODBC Data Source Administrator and WMS support multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

1. Select **Start→All Programs→HP ODBC 2.0→MS ODBC Administrator**.

   The ODBC Data Source Administrator dialog box appears:

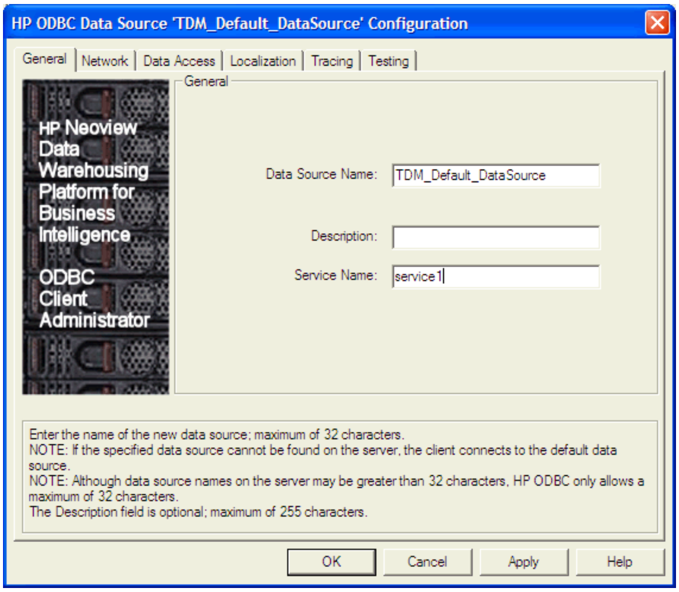


User Data Sources:

Name	Driver
dBASE Files	Microsoft dBase Driver (*.dbf)
Excel Files	Microsoft Excel Driver (*.xls)
MS Access Database	Microsoft Access Driver (*.mdb)
TDM_Default_DataSource	HP ODBC 2.0
Visio Database Samples	Microsoft Access Driver (*.MDB)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

2. Under the User DSN tab, select the Data Source for which you want to specify a service.
3. Click **Configure...**.
4. Enter a name in the Service Name box:



HP ODBC Data Source 'TDM_Default_DataSource' Configuration
General
Network
Data Access
Localization
Tracing
Testing
General
HP Neoview Data Warehousing Platform for Business Intelligence
ODBC Client Administrator
OK
Cancel
Apply
Help

Data Source Name: TDM_Default_DataSource

Description:

Service Name: service1

Enter the name of the new data source; maximum of 32 characters.
NOTE: If the specified data source cannot be found on the server, the client connects to the default data source.
NOTE: Although data source names on the server may be greater than 32 characters, HP ODBC only allows a maximum of 32 characters.
The Description field is optional; maximum of 255 characters.

5. Click **OK** to accept the changes.

The service name that you set in the ODBC Administrator applies to queries that run in the data source. A service name that is set in a JDBC or ODBC client application or in an NCI session takes precedence over the setting in the ODBC Administrator.

Setting the Service in the MXODSN File on a Linux or UNIX Workstation

On a Linux or UNIX workstation, you can set the service name for a client data source in the MXODSN file. Add a `ServiceName` entry at the DS level as shown in **boldface** below:

```
[TDM_Default_DataSource]
Description                   = Default Data Source
Catalog                       = CAT
Schema                        = SCH
DataLang                      = 0
FetchBufferSize               = SYSTEM_DEFAULT
Server                        = TCP:1.2.3.4:18650
SQL_ATTR_CONNECTION_TIMEOUT = SYSTEM_DEFAULT
SQL_LOGIN_TIMEOUT             = SYSTEM_DEFAULT
SQL_QUERY_TIMEOUT             = NO_TIMEOUT
ServiceName                   = MY_SERVICE
ReplacementCharacter          = ?
```

The service name that you set in the MXODSN file applies to queries that run in the data source. A service name that is set in a JDBC or ODBC client application or in an NCI session takes precedence over the setting in the MXODSN file.

Using the SQLSetConnectAttr() Function to Associate Queries With a Service

In an ODBC application, you can use the `SQLSetConnectAttr()` function to set a service for the queries in the application. In the `SQLSetConnectAttr()` function, set the attribute argument to `1030` and the value pointer to the service name:

```
SQLSetConnectAttr( hdbc, 1030, "salesqueries",  SQL_NTS);
```

In the previous example, WMS assigns all subsequent queries in the ODBC application to the SALESQUERIES service.

The `SQLSetConnectAttr()` setting takes precedence over the service-name setting in the client data source. However, a SET SERVICE setting in the application takes precedence over the `SQLSetConnectAttr()` setting.

To get the service name that is in effect for an ODBC application, use the `SQLGetConnectAttr()` function and set the attribute argument to `1030`, the value pointer to a variable for the service name, and the buffer length to 128:

```
SQLGetConnectAttr( hdbc, 1030, servicename, 128, SQL_NTS);
```

For more information about using the ODBC driver, see the *Neoview ODBC Drivers Manual*.

Using the setServiceName() Method to Associate Queries With a Service

In a JDBC application, you can use the `setServiceName()` method of the `com.hp.t4jdbc.HPT4Connection` class to set a service for the queries in the application:

```
setServiceName("salesqueries");
```

In the previous example, WMS assigns all subsequent queries in the JDBC application to the SALESQUERIES service.

The `setServiceName()` setting takes precedence over the service-name setting in the client data source. However, a SET SERVICE setting in the application takes precedence over the `setServiceName()` setting.

To get the service name that is in effect for a JDBC application, use the `getServiceName()` method of the `com.hp.t4jdbc.HPT4Connection` class:

```
String serviceName = getServiceName();
```

For more information about using the JDBC Type 4 driver, see the *Neoview JDBC Type 4 Driver Programmer's Reference*.

Using the SET SERVICE Statement to Associate Queries With a Service

SET SERVICE Statement

In NCI or in an ODBC or JDBC client application, use the SET SERVICE statement to associate SQL queries with a service.

In NCI, in SQL mode only, you can enter SET SERVICE interactively in the command-line interface, or you can put the SET SERVICE statement in a script file that you run in NCI. For example, this SET SERVICE statement sets the service name to DATALOADING in a script file that you run in NCI:

```
set service dataloading;

create schema persnl;
create table employee...
```

IMPORTANT: If the service name contains multibyte characters, enclose the service name in double quotes in the SET SERVICE statement to ensure correct translation. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

If the service that you specify does not exist, WMS first looks for a service name that matches the second part of the client role, such as MGR in ROLE.MGR. If WMS finds a service name that matches the role, WMS manages the queries in that service. If WMS does not find a service name that matches the role, WMS manages queries in the default service, HP_DEFAULT_SERVICE.

In an ODBC or JDBC client application, you can use the SET SERVICE statement as you would any other SQL SET statement. This example shows SET SERVICE in an ODBC application:

```
sprintf(stmt, "SET SERVICE %s", servicename);
retcode = SQLExecDirect(hstmt, stmt, SQL_NTS);
```

This example shows SET SERVICE in a JDBC application:

```
Connection conn = DriverManager.getConnection("jdbc:default:connection");
Statement stmt = conn.createStatement();
stmt.execute("set service dataloading");
```

In a client application, the SET SERVICE statement takes precedence over a `SQLSetConnectAttr()` or `setServiceName()` setting.

For more information about SET SERVICE, see the *Neoview SQL Reference Manual*.

GET SERVICE Statement

The GET SERVICE statement shows the service that is in effect for the current session and whether the SQL plan or SQL text are enabled for that service. You can use GET SERVICE in NCI or in an ODBC or JDBC client application. For example, in NCI:

```
SQL>get service;

CUR_SERVICE                                        PLAN        TEXT
-------------------------------------------------- ----------- -----------
HP_DEFAULT_SERVICE                                           0           0

--- SQL operation complete.

SQL>
```

If you did not explicitly set a service for the session, GET SERVICE shows that the default service, HP_DEFAULT_SERVICE, is in effect:

```
SQL>get service;

CUR_SERVICE                                        PLAN        TEXT
-------------------------------------------------- ----------- -----------
HP_DEFAULT_SERVICE                                           0           0

--- SQL operation complete.

SQL>
```

For more information about GET SERVICE, see the *Neoview SQL Reference Manual*.

5 Managing Workload in WMS

Monitoring System Resources

Using the STATUS WMS Command to Monitor System Resources

In NCI, this STATUS WMS command displays information about the state of the system and the system resources:

```
NS%status wms;

STATE     MAX_CPU_BUSY MAX_MEM_USAGE STATS_INTERVAL DISK_IO                  DISK_CACHE               CPU_BUSY
                 MEM_USAGE
--------- ------------ ------------- -------------- ------------------------ ------------------------
------------------------ ------------------------
ACTIVE               0             0              5                    42.08                    17.46
            0.99                    87.92

--- WMS operation complete.

NS%
```

For more information, see the “STATUS Command” (page 66).

Monitoring Services and Queries

Displaying the Status of Services

Use the STATUS SERVICE command to display the status of services in WMS. The STATUS SERVICE command displays this status information about the existing services in WMS:

SERVICE_NAME	Name of the service
STATE	Activity state of the service
SERVICE_PRIORITY	The priority of the service relative to other services
MAX_CPU_BUSY	A percentage from zero to 100 of the total CPU busy allowed for the service. This value is an average of CPU busy across all segments in the cluster. The default is zero, meaning that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE	A percentage from zero to 100 of memory usage allowed for the service. This value is an average of memory usage across all segments in the cluster. The default is zero, meaning that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.

ACTIVE_TIME	The time period during which the service is active and can execute queries. The default is 00:00 through 23:59.
SQL_PLAN	Displays PLAN to indicate that the query execution plans are collected for display or NO_PLAN to indicate that no plans are collected for display
SQL_TEXT	Displays TEXT to indicate that the SQL syntax of executing queries is collected for display or NO_TEXT to indicate that no SQL text is collected for display
EXEC_TIMEOUT	The number of minutes a query is allowed to remain in the executing state before WMS cancels the query. The default is zero, meaning no timeout.
WAIT_TIMEOUT	The number of minutes a query is allowed to remain in the waiting state before WMS cancels the query. The default is zero, meaning no timeout.

A service is in the HOLD state if the current time is outside the active time range of the service or if a user holds the service by issuing a HOLD command. A service is in the ACTIVE state if the current time is within the active range of the service or if a user releases the service from the HOLD state. For more information, see the ACTIVE_TIME threshold in “Adding a Service” (page 29), or see “Holding a Service” (page 51) or “Releasing a Service” (page 51).

For example, this STATUS SERVICE ALL command displays information about the state and thresholds of all services being managed by WMS:

```
NS%status service all;

SERVICE_NAME                                                                                     STATE
SERVICE_PRIORITY MAX_CPU_BUSY MAX_MEM_USAGE ACTIVE_TIME         SQL_PLAN SQL_TEXT EXEC_TIMEOUT WAIT_TIMEOUT
------------------------------------------------------------------------------------------------ --------
---------------- ------------ ------------- ------------------  -------- -------- ------------ -----------
SERVICE1                                                                                         ACTIVE   HIGH
                        0             0 FROM 00:00 TO 23:59 PLAN     TEXT                0            0
DATALOADING                                                                                      HOLD
MEDIUM-HIGH                 0             0 FROM 22:00 TO 23:59 NO_PLAN  NO_TEXT             0            0
SERVICE2                                                                                       ACTIVE   MEDIUM
                      0             0 FROM 00:00 TO 23:59 PLAN     TEXT                0            0
SALESQUERIES                                                                                     HOLD
LOW-MEDIUM                  0             0 FROM 08:00 TO 10:00 NO_PLAN  NO_TEXT             0            0
HP_DEFAULT_SERVICE                                                                               ACTIVE   LOW
                        0            89 FROM 00:00 TO 23:59 PLAN     TEXT                0            0
SERVICE3                                                                                         ACTIVE   LOW
                        0             0 FROM 00:00 TO 23:59 PLAN     TEXT                0            0

--- WMS operation complete.

NS%
```

For example, this STATUS SERVICE <service> command displays information about the state and thresholds of a specified service being managed by WMS:

```
NS%status service hp_default_service;

SERVICE_NAME                                                                                     STATE
SERVICE_PRIORITY MAX_CPU_BUSY MAX_MEM_USAGE ACTIVE_TIME         SQL_PLAN SQL_TEXT EXEC_TIMEOUT WAIT_TIMEOUT
------------------------------------------------------------------------------------------------ --------
---------------- ------------ ------------- ------------------  -------- -------- ------------ -----------
HP_DEFAULT_SERVICE                                                                               ACTIVE   LOW
                        0            89 FROM 00:00 TO 23:59 PLAN     TEXT                0            0

--- WMS operation complete.

NS%
```

For example, this STATUS SERVICE STATS command displays statistics for all the services on the system:

```
NS%status service stats;

SEGMENT     SERVICE_NAME
STATE       EXECUTING   WAITING     HOLDING
----------- ------------------------------------------------------------------------------------------------
----------- ----------- ----------- -----------
          1 SERVICE1
ACTIVE                0           0           0
          2 SERVICE1
ACTIVE                1           0           0
          3 SERVICE1
ACTIVE                1           0           0
          1 DATALOADING
HOLD                  0           0           0
          2 DATALOADING
HOLD                  0           0           0
```

```
          3 DATALOADING
HOLD                  0           0           0
          1 SERVICE2
ACTIVE                1           0           0
          2 SERVICE2
ACTIVE                1           0           0
          3 SERVICE2
ACTIVE                0           0           0
          1 SALESQUERIES
HOLD                  0           0           0
          2 SALESQUERIES
HOLD                  0           0           0
          3 SALESQUERIES
HOLD                  0           0           0
          1 SERVICE3
ACTIVE                1           0           0
          2 SERVICE3
ACTIVE                0           0           0
          3 SERVICE3
ACTIVE                1           0           0
          1 HP_DEFAULT_SERVICE
ACTIVE                0           0           0
          2 HP_DEFAULT_SERVICE
ACTIVE                0           0           0
          3 HP_DEFAULT_SERVICE
ACTIVE                0           0           0

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Displaying the Status and Compile-Time Statistics of All Queries

Use the STATUS QUERIES ALL command to display the status of all queries being managed by WMS. For example:

```
NS%status queries all;

QUERY_ID
                                                   QUERY_STATE SERVICE_NAME
                                                 EST_COST                 EST_CPU_TIME             EST_IO_TIME
              EST_MSG_TIME             EST_IDLE_TIME            EST_TOTAL_TIME           EST_CARDINALITY
    PROCESS_NAME                     QUERY_NAME
                                                                                        ROLE_NAME
                             SERVICE_INDEX QUERY_INDEX START_TS                   ENTRY_TS
ENTRY_JTS            LAST_UPDATED
--------------------------------------------------------------------------------------------------------------
 ----------- ------------------------------------------------------------------------------------------------
------------------------ ------------------------ ------------------------ ------------------------
------------------------ ------------------------ ------------------------ --------------------------------
--------------------------------------------------------------------------------------------------------------
 -------------------------------------------------- ------------- ----------- --------------------------
-------------------------- -------------------- --------------------
MXID01001010517212075562920126981000000004004DBA100_238_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405125845796156       2.4270030100716777
0.43943207564842224       0.4502995243688986      0.39207400850312774       2.1405125845796156
      1.0 \WMS0101.$Z12K                   PC-D530||FASTJDBC
                                                                                      DBA1

                                  524312           0 2008-04-03 18:18:13.372013 2008-04-03 18:18:13.372035
212075561893372035   212075561898350133
MXID01002000687212075563079285896000000012114SUPER.SERVICES00_7794_SQL_CUR_3
                                                   COMPLETED   HP_DEFAULT_SERVICE
                                                       0.03621300703690157        0.107988008066793
0.11072000134299742      0.11150500165967969      0.09800000134418951      0.03621300703690157
    100.0 \WMS0102.$Z0DT                   CACNARAIN||COAST.exe
                                                                                              SUPER.SERVICES
                                       262168          1 2008-04-03 18:17:48.675194 2008-04-03 18:17:48.675225
   212075561868675225   212075561895388641
MXID01003020448212075562925137594000000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FJ                   PC-D530||FASTJDBC
                                                                                      DBA1

                                  458776           0 2008-04-03 18:18:11.032345 2008-04-03 18:18:11.032383
212075561891032383   212075561895981400
MXID01001021072212075562925165799000000004104DBA100_239_S1
                                                   EXECUTING   SERVICE3
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0101.$Z12N                   PC-D530||FASTJDBC
                                                                                      DBA1

                                  589848           1 2008-04-03 18:18:18.350083 2008-04-03 18:18:18.350125
212075561898350125   212075561898350133
MXID01002020452212075562925258497000000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
```

```
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DR                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:09.613894 2008-04-03 18:18:09.613916
212075561889613916   212075561895388641
MXID01003010491212075562925097818000000004004DBA100_234_S1
                                                   EXECUTING   SERVICE3
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FH                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  589848           1 2008-04-03 18:18:15.981358 2008-04-03 18:18:15.981392
212075561895981392   212075561895981400
MXID01002000687212075563079285896000000012114SUPER.SERVICES00_7802_SQL_CUR_3
                                                   COMPLETED   HP_DEFAULT_SERVICE
                                                      0.017414606623438217      0.10269460796881982
0.11072000134299742      0.09800000134418951      0.09800000134418951     0.017414606623438217
    100.0 \WMS0102.$Z0DT                   CACNARAIN||COAST.exe
                                                                                              SUPER.SERVICES
                                       262168          2 2008-04-03 18:18:13.561928 2008-04-03 18:18:13.561968
   212075561893561968   212075561895388641
MXID01002010476212075562920218543000000004004DBA100_234_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DN                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312           3 2008-04-03 18:18:15.388603 2008-04-03 18:18:15.388633
212075561895388633   212075561895388641

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Displaying the Status and Compile-Time Statistics of Queries in a Service

Use the STATUS QUERIES SERVICE command to display the status of queries in a service. For example:

```
NS%status queries service service1;

QUERY_ID
                                                   QUERY_STATE SERVICE_NAME
                                                 EST_COST                 EST_CPU_TIME             EST_IO_TIME
              EST_MSG_TIME             EST_IDLE_TIME            EST_TOTAL_TIME           EST_CARDINALITY
    PROCESS_NAME                     QUERY_NAME
                                                                                        ROLE_NAME
                             SERVICE_INDEX QUERY_INDEX START_TS                   ENTRY_TS
ENTRY_JTS            LAST_UPDATED
-----------------------------------------------------------------------------------------------------------------
 ----------- -------------------------------------------------------------------------------------------------
------------------------ ------------------------ ------------------------  ------------------------
------------------------ ------------------------ ------------------------  --------------------------------
-----------------------------------------------------------------------------------------------------------------
 -------------------------------------------------- ------------- ----------- --------------------------
-------------------------- -------------------- --------------------
MXID01002020452212075562925258497000000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DR                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:09.613894 2008-04-03 18:18:09.613916
212075561889613916   212075561895388641
MXID01003020448212075562925137594000000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FJ                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:11.032345 2008-04-03 18:18:11.032383
212075561891032383   212075561895981400

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Filtering the Status and Compile-Time Statistics of Queries

You can use different STATUS commands to filter query status results according to different criteria:

- Use the STATUS QUERIES <state> command to filter query status results according to the query state (for example, waiting, executing, or holding).
- Use the STATUS QUERIES SERVICE <service> <state> command to filter query status results according to the service and optionally the query state (for example, waiting, executing, or holding).
- Use the STATUS QUERIES ROLE <name> <state> command to filter query status results according to the client role or alias, and optionally the query state (for example, waiting, executing, or holding).
- Use the STATUS QUERY <query-id> command to filter query status results according to the query ID.

For example, this STATUS QUERIES <state> command displays status information about queries in the executing state:

```
NS%status queries executing;

QUERY_ID
                                                   QUERY_STATE SERVICE_NAME
                                                 EST_COST                 EST_CPU_TIME             EST_IO_TIME
              EST_MSG_TIME             EST_IDLE_TIME            EST_TOTAL_TIME           EST_CARDINALITY
    PROCESS_NAME                     QUERY_NAME
                                                                                        ROLE_NAME
                             SERVICE_INDEX QUERY_INDEX START_TS                   ENTRY_TS
ENTRY_JTS            LAST_UPDATED
--------------------------------------------------------------------------------------------------------------
 ----------- ------------------------------------------------------------------------------------------------
------------------------ ------------------------ ------------------------ ------------------------
------------------------ ------------------------ ------------------------ --------------------------------
--------------------------------------------------------------------------------------------------------------
 -------------------------------------------------- ------------- ----------- --------------------------
------------------------- -------------------- ---------------------
MXID0100101051721207556292012698100000004004DBA100_238_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405125845796156       2.4270030100716777
0.43943207564842224       0.4502995243688986      0.39207400850312774       2.1405125845796156
      1.0 \WMS0101.$Z12K                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312           0 2008-04-03 18:18:13.372013 2008-04-03 18:18:13.372035
212075561893372035   212075561898350133
MXID0100202045221207556292525849700000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DR                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:09.613894 2008-04-03 18:18:09.613916
212075561889613916   212075561895388641
MXID0100302044821207556292513759400000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FJ                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:11.032345 2008-04-03 18:18:11.032383
212075561891032383   212075561895981400
MXID0100102107221207556292516579900000004104DBA100_239_S1
                                                   EXECUTING   SERVICE3
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0101.$Z12N                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  589848           1 2008-04-03 18:18:18.350083 2008-04-03 18:18:18.350125
212075561898350125   212075561898350133
MXID0100201047621207556292021854300000004004DBA100_234_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DN                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312           3 2008-04-03 18:18:15.388603 2008-04-03 18:18:15.388633
212075561895388633   212075561895388641
MXID0100301049121207556292509781800000004004DBA100_234_S1
                                                   EXECUTING   SERVICE3
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FH                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  589848           1 2008-04-03 18:18:15.981358 2008-04-03 18:18:15.981392
212075561895981392   212075561895981400

--- WMS operation complete.

NS%
```

For example, this STATUS QUERIES SERVICE command displays status information about queries in the SERVICE2 service that are in the executing state:

```
NS%status queries service service2 executing;

QUERY_ID
                                                   QUERY_STATE SERVICE_NAME
                                                 EST_COST                 EST_CPU_TIME             EST_IO_TIME
              EST_MSG_TIME             EST_IDLE_TIME           EST_TOTAL_TIME           EST_CARDINALITY
    PROCESS_NAME                     QUERY_NAME
                                                                                        ROLE_NAME
                             SERVICE_INDEX QUERY_INDEX START_TS                   ENTRY_TS
ENTRY_JTS            LAST_UPDATED
--------------------------------------------------------------------------------------------------------------
 ----------- ------------------------------------------------------------------------------------------------
------------------------ ------------------------ ------------------------ ------------------------
------------------------ ------------------------ ------------------------ --------------------------------
--------------------------------------------------------------------------------------------------------------
 -------------------------------------------------- ------------ ----------- --------------------------
------------------------ -------------------- --------------------
MXID01001010517212075562920126981000000004004DBA100_238_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405125845796156       2.4270030100716777
0.43943207564842224       0.4502995243688986      0.39207400850312774       2.1405125845796156
      1.0 \WMS0101.$Z12K                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312           0 2008-04-03 18:18:13.372013 2008-04-03 18:18:13.372035
212075561893372035   212075561898350133
MXID01002010476212075562920218543000000004004DBA100_234_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DN                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312           3 2008-04-03 18:18:15.388603 2008-04-03 18:18:15.388633
212075561895388633   212075563928780174

--- WMS operation complete.

NS%
```

For example, this STATUS QUERIES ROLE command displays status information about queries associated with the client alias, DBA1:

```
NS%status queries role dba1;

QUERY_ID
                                                   QUERY_STATE SERVICE_NAME
                                                 EST_COST                 EST_CPU_TIME             EST_IO_TIME
              EST_MSG_TIME             EST_IDLE_TIME           EST_TOTAL_TIME           EST_CARDINALITY
    PROCESS_NAME                     QUERY_NAME
                                                                                        ROLE_NAME
                             SERVICE_INDEX QUERY_INDEX START_TS                   ENTRY_TS
ENTRY_JTS            LAST_UPDATED
--------------------------------------------------------------------------------------------------------------
 ----------- ------------------------------------------------------------------------------------------------
------------------------ ------------------------ ------------------------ ------------------------
------------------------ ------------------------ ------------------------ --------------------------------
--------------------------------------------------------------------------------------------------------------
 -------------------------------------------------- ------------ ----------- --------------------------
------------------------ -------------------- --------------------
MXID01001010517212075562920126981000000004004DBA100_238_S1
                                                   EXECUTING   SERVICE2
                                                        2.1405125845796156       2.4270030100716777
0.43943207564842224       0.4502995243688986      0.39207400850312774       2.1405125845796156
      1.0 \WMS0101.$Z12K                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312           0 2008-04-03 18:18:13.372013 2008-04-03 18:18:13.372035
212075561893372035   212075561898350133
MXID01002020452212075562925258497000000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DR                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:09.613894 2008-04-03 18:18:09.613916
212075561889613916   212075563928780174
MXID01003020448212075562925137594000000004204DBA100_244_S1
                                                   EXECUTING   SERVICE1
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FJ                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  458776           0 2008-04-03 18:18:11.032345 2008-04-03 18:18:11.032383
212075561891032383   212075561895981400
MXID01001021072212075562925165799000000004104DBA100_239_S1
                                                   EXECUTING   SERVICE3
                                                        2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0101.$Z12N                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  589848          1 2008-04-03 18:18:18.350083 2008-04-03 18:18:18.350125
212075561898350125   212075561898350133
MXID01002010476212075562920218543000000004004DBA100_234_S1
                                                   EXECUTING   SERVICE2
                                                         2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DN                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  524312          3 2008-04-03 18:18:15.388603 2008-04-03 18:18:15.388633
212075561895388633   212075563928780174
MXID01003010491212075562925097818000000004004DBA100_234_S1
                                                   EXECUTING   SERVICE3
                                                         2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0103.$Z0FH                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  589848          1 2008-04-03 18:18:15.981358 2008-04-03 18:18:15.981392
212075561895981392   212075561895981400

--- WMS operation complete.

NS%
```

For example, this STATUS QUERY <query-id> command displays status information about a query identified by the specified query ID:

```
NS%status query MXID01002010476212075562920218543000000004004DBA100_234_S1;

QUERY_ID
                                                   QUERY_STATE SERVICE_NAME
                                                 EST_COST                 EST_CPU_TIME             EST_IO_TIME
              EST_MSG_TIME             EST_IDLE_TIME            EST_TOTAL_TIME           EST_CARDINALITY
    PROCESS_NAME                     QUERY_NAME
                                                                                         ROLE_NAME
                             SERVICE_INDEX QUERY_INDEX START_TS                   ENTRY_TS
ENTRY_JTS            LAST_UPDATED
---------------------------------------------------------------------------------------------------------------
 ----------- -------------------------------------------------------------------------------------------------
------------------------ ------------------------ ------------------------ ------------------------
------------------------ ------------------------ ------------------------ --------------------------------
---------------------------------------------------------------------------------------------------------------
 -------------------------------------------------- ------------- ----------- --------------------------
-------------------------- -------------------- --------------------
MXID01002010476212075562920218543000000004004DBA100_234_S1
                                                   EXECUTING   SERVICE2
                                                         2.1405113621293412       2.4270026755241196
0.43943207564842224      0.45029863646618223      0.39207400850312774       2.1405113621293412
      1.0 \WMS0102.$Z0DN                   PC-D530||FASTJDBC
                                                                                      DBA1
                                  536476          3 2008-04-03 18:18:15.388603 2008-04-03 18:18:15.388633
212075561895388633   212075561927651228

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Displaying the Total Number of Executing Queries

Use the STATUS QUERIES EXECUTING STATS command to display the total number of executing queries in various states on each segment of the Neoview platform.

For example, this STATUS QUERIES EXECUTING STATS command displays the total number of executing queries in various states per segment:

```
NS%status queries executing stats;

SEGMENT     TOT_INITIAL TOT_OPEN    TOT_EOF     TOT_CLOSE   TOT_DEALLOCATED TOT_FETCH   TOT_CLOSE_TABLES
TOT_PREPARE
----------- ----------- ----------- ----------- ----------- --------------- ----------- ----------------
-----------
          1           2           0           0           0               0           0                0
     0
          2           2           0           0           0               0           0                0
     0
          3           2           0           0           0               0           0                0
     0

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Displaying Runtime Statistics of All Queries

Use the STATUS QUERIES ALL STATS or STATUS QUERY ALL STATS command to display the runtime statistics of all queries being managed by WMS.

NOTE: By default, WMS collects runtime statistics for all executing queries at an interval of five seconds.

For example, this STATUS QUERIES ALL STATS command displays the runtime statistics of all queries being managed by WMS:

```
NS%status queries all stats;

QUERY_ID
                                                   QUERY_STATE STATEMENT_ID

    STATEMENT_TYPE        COMP_START_TIME      COMP_END_TIME        EXEC_START_TIME      EXEC_END_TIME
EXEC_STATE            ELAPSED_TIME         ACCESSED_ROWS        USED_ROWS            MESSAGE_COUNT
MESSAGE_BYTES        STATS_BYTES          DISK_IOS             LOCK_WAITS           LOCK_ESCALATIONS
PROCESS_BUSYTIME    OPENS               OPEN_TIME          SQL_ERROR_CODE NUM_ROWS_IUD        STATS_ERROR_CODE
 SQL_SPACE_ALLOC SQL_SPACE_USED SQL_HEAP_ALLOC SQL_HEAP_USED SQL_CPU_TIME       EID_SPACE_ALLOC EID_SPACE_USED
 EID_HEAP_ALLOC EID_HEAP_USED EST_ACCESSED_ROWS        EST_USED_ROWS            PROCESSES_CREATED
PROCESS_CREATE_TIME  LAST_UPDATED
--------------------------------------------------------------------------------------------------------------
 -----------
--------------------------------------------------------------------------------------------------------------
 --------------------- -------------------- -------------------- -------------------- --------------------
--------------------- -------------------- -------------------- -------------------- --------------------
-------------------- -------------------- -------------------- -------------------- --------------------
------------------- ------------------- ------------------- -------------- ------------------- ----------------
 --------------- -------------- -------------- ------------- ------------------- --------------- --------------
 -------------- ------------- ------------------------ ------------------------ --------------------
-------------------- --------------------
MXID01001010517212075562920126981000000004004DBA100_238_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0          -8921                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561898449906
MXID010020006872120755630792858960000000012114SUPER.SERVICES00_7794_SQL_CUR_3
                                                   COMPLETED

    SELECT_NON_UNIQUE       212075587066424882   212075587068654585   212075587068676346   212075587069616089
DEALLOCATED                         939743                    1                    1                    2
           12048                  344                    0                    0                    0
        259                    1                  374              0                   -1                0
          96             65              7             5                   78              64             55
           97             1                    100.0                    100.0                    0
        0   212075561895388651
MXID01003020448212075562925137594000000004204DBA100_244_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0          -8921                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561897764939
MXID01001021072212075562925165799000000004104DBA100_239_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0              0                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561898449906
MXID01002020452212075562925258497000000004204DBA100_244_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0          -8921                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561895388651
MXID01003010491212075562925097818000000004004DBA100_234_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
```

```
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0              0                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561897764939
MXID01002000687212075563079285896000000012114SUPER.SERVICES00_7802_SQL_CUR_3
                                                   COMPLETED

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0              0                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561895388651
MXID01002010476212075562920218543000000004004DBA100_234_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE                        0                    0                    0                   -1
INITIAL                                 -1                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0              0                    0                0
           0              0              0             0                    0               0              0
            0             0                      0.0                      0.0                    0
        0   212075561895388651

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Displaying Runtime Statistics of a Specific Query

Use the STATUS QUERY <query-id> STATS command to display the runtime statistics of a query being managed by WMS.

NOTE: By default, WMS collects runtime statistics for all executing queries at an interval of five seconds.

For example, this STATUS QUERY <query-id> STATS command displays the runtime statistics of a specific query being managed by WMS:

```
NS%status query MXID01002010476212075562920218543000000004004DBA100_234_S1 stats;

QUERY_ID
                                                   QUERY_STATE STATEMENT_ID

    STATEMENT_TYPE        COMP_START_TIME      COMP_END_TIME        EXEC_START_TIME      EXEC_END_TIME
EXEC_STATE            ELAPSED_TIME         ACCESSED_ROWS        USED_ROWS            MESSAGE_COUNT
MESSAGE_BYTES        STATS_BYTES          DISK_IOS             LOCK_WAITS           LOCK_ESCALATIONS
PROCESS_BUSYTIME    OPENS              OPEN_TIME           SQL_ERROR_CODE NUM_ROWS_IUD        STATS_ERROR_CODE
 SQL_SPACE_ALLOC SQL_SPACE_USED SQL_HEAP_ALLOC SQL_HEAP_USED SQL_CPU_TIME       EID_SPACE_ALLOC EID_SPACE_USED
 EID_HEAP_ALLOC EID_HEAP_USED EST_ACCESSED_ROWS        EST_USED_ROWS            PROCESSES_CREATED
PROCESS_CREATE_TIME  LAST_UPDATED
--------------------------------------------------------------------------------------------------------------
 -----------
--------------------------------------------------------------------------------------------------------------
 --------------------- -------------------- -------------------- -------------------- --------------------
--------------------- -------------------- -------------------- -------------------- --------------------
-------------------- -------------------- -------------------- -------------------- --------------------
------------------- ------------------- ------------------- ------------- ------------------- ----------------
 --------------- -------------- -------------- ------------- ------------------- -------------- --------------
 -------------- ------------- ------------------------ ------------------------ --------------------
-------------------- --------------------
MXID01002010476212075562920218543000000004004DBA100_234_S1
                                                   EXECUTING

    SELECT_NON_UNIQUE       212075587094687749   212075587095358574   212075587095389714                   -1
OPEN                              64941004                    0                    0                    0
               0                    0                    0                    0                    0
          0                    0                    0              0                   -1                0
         105             86             15             9                  603               0              0
            0             0                    624.0                    624.0                    8
    25498   212075561960330724

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Displaying the Query Explain Plan

Use the STATUS QUERY <query-id> PLAN command to display the execution plan of a specific query. For the STATUS QUERY <query-id> PLAN command to succeed, the output option, PLAN, must be enabled for the service. If the default NO_PLAN is specified for the service, the STATUS QUERY <query-id> PLAN command does not return a query execution plan.

For example, this STATUS QUERY <query-id> PLAN command displays the query execution plan for a specified query ID:

```
NS%status query MXID01002010476212075562920218543000000004004DBA100_234_S1 plan;

QUERY_PLAN

--------------------------------------------------------------------------------------------------------------

LC RC OP OPERATOR OPT DESCRIPTION CARD
---- ---- ---- -------------------- -------- -------------------- ---------

18 . 19 root 1.00E+000
17 . 18 sort_parti
al_aggr_ro 1.00E+000
16 . 17 esp_exchange 1:4(hash2) 1.00E+000
15 . 16 sort_partial_aggr_le 1.00E+000
14 3 15 hybrid_hash_join 5.92E+008
13 6 14 hybrid_hash_joi
n 3.79E+006
12 9 13 hybrid_hash_join 2.43E+004
11 . 12 esp_exchange 4(hash2):1 1.56E+002
10 . 11 partition_access 1.56E+002
. . 10 file_scan fs fr COLS 1.56E+00
2
8 . 9 esp_exchange 4(rep-b):1 1.56E+002
7 . 8 partition_access 1.56E+002
. . 7 file_scan fs fr COLS 1.56E+002
5 . 6 esp_exchange 4(rep-b):1 1.56E+002
4 . 5 pa
rtition_access 1.56E+002
. . 4 file_scan fs fr COLS 1.56E+002
2 . 3 esp_exchange 4(rep-b):1 1.56E+002
1 . 2 partition_access 1.56E+002
. . 1 file_scan fs fr COL
S 1.56E+002


--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

To interpret explain plans, see the *Neoview Query Guide* and the *Neoview Query Support Guide*.

Displaying the Query SQL Text

Use the STATUS QUERY <query-id> TEXT command to display the SQL text of a specific query. For the STATUS QUERY <query-id> TEXT command to succeed, the output option, TEXT, must be enabled for the service. If NO_TEXT, the default, is specified for the service, the STATUS QUERY <query-id> TEXT command fails to return SQL text for a query.

For example, this STATUS QUERY <query-id> TEXT command displays the SQL text for a specified query ID:

```
NS%status query MXID01001000483212066839774858733000000285214SUPER.SERVICES00_13391_S1 text;

QUERY_TEXT

--------------------------------------------------------------------------------------------------------------
select od.ordernum, sum(qty_ordered * price)
from sales.parts p, sales.odetail od
where od.partnum = p.partnum and od.ordernum in
  (select o.ordernum
   from sales.orders o, sales.customer c
   where o.custnum = c.custnum and state = 'CALIFORNIA')
group by od.ordernum;

--- WMS operation complete.

NS%
```

For the syntax, see the “STATUS Command” (page 66).

Managing Services

Holding All Services

When you hold all services in WMS, you prevent the waiting queries associated with all the services from being executed. While the services are on hold, WMS allows queries that are currently executing in the services to finish executing, but WMS rejects new incoming queries associated with the services.

Use the HOLD SERVICE ALL command to hold all services in WMS. For example:

```
NS%hold service all;

--- WMS operation complete.

NS%
```

For the syntax, see the "HOLD Command" (page 63).

Holding a Service

When you hold a service in WMS, you prevent the waiting queries associated with the service from being executed. While the service is on hold, WMS allows queries that are currently executing in the service to finish executing, but WMS rejects new incoming queries associated with the service.

Use the HOLD SERVICE command to hold the waiting queries of a specific service in WMS. For example, this HOLD command holds all waiting queries in the DATALOADING service:

```
NS%hold service dataloading;

--- WMS operation complete.

NS%
```

For the syntax, see the "HOLD Command" (page 63).

Releasing All Services

To release all services that are currently being held in WMS, use the RELEASE SERVICE ALL command. For example:

```
NS%release service all;

--- WMS operation complete.

NS%
```

For the syntax, see the "RELEASE Command" (page 66).

Releasing a Service

Use the RELEASE SERVICE command to release a service and all its queries, which are being held in WMS. For example, this RELEASE command releases all holding queries in the DATALOADING service:

```
NS%release service dataloading;

--- WMS operation complete.

NS%
```

For the syntax, see the “RELEASE Command” (page 66).

Managing Queries

You can manage queries in WMS by using the HOLD, RELEASE, and CANCEL commands. The HOLD command enables you to hold a query that is in the waiting queue, and the RELEASE command enables you to release a holding query and put it back into the waiting queue. You can also kill a waiting or executing query to free system resources by using the CANCEL command. For more information, see:

- “Holding a Query” (page 52)
- “Releasing a Query” (page 52)
- “Cancelling a Query” (page 52)

Queries in WMS are either waiting to execute or executing. A client application that issues a query has no way of knowing whether the query is in the waiting queue. The client application perceives the time spent by the query in the waiting queue as part of the execution time. To determine whether a particular query is in the waiting queue, use the STATUS QUERY <query-id> command. For more information, see “Filtering the Status and Compile-Time Statistics of Queries” (page 44) or the “STATUS Command” (page 66).

Holding a Query

Use the HOLD command to hold a query in the waiting queue. The HOLD command takes the query out of the waiting queue, thus allowing other queries in the waiting queue to proceed to the executing state when system resources become available. You can put the query back into the waiting queue by issuing a RELEASE command. For more information, see “Releasing a Query” (page 52).

NOTE: To hold a query, the query must be in the waiting queue. You cannot hold an executing query. To cancel an executing query, see “Cancelling a Query” (page 52).

For example, this HOLD command holds a specific waiting query:

```
NS%hold query MXID01001120500212040077432400244000000000205susan00_37_STMT1
+>;

--- WMS operation complete.

NS%
```

For the syntax, see the “HOLD Command” (page 63).

Releasing a Query

Use the RELEASE QUERY command to release a query that is being held in WMS and put it back into the waiting queue. For example, this RELEASE command releases a holding query:

```
NS%release query MXID01001120500212040077432400244000000000205susan00_37_STMT1
+>;

--- WMS operation complete.

NS%
```

For the syntax, see the “RELEASE Command” (page 66).

Cancelling a Query

Use the CANCEL QUERY command to cancel an executing or waiting query in WMS and free system resources. For example, this CANCEL command kills a specific executing query:

```
NS%cancel query MXID01001120500212040077432400244000000000205susan00_37_STMT1
+>;
```

```
--- WMS operation complete.

NS%
```

For the syntax, see the “CANCEL Command” (page 61).

When you cancel a query, you kill the NDCS server that is handling the query.

If you cancel an executing query from a JDBC client application, including NCI, WMS returns this message to the client application:

```
There was a problem reading from the server
The message header was not long enough
```

If you cancel an executing query from an ODBC client application, WMS returns this message about a communication link failure to the client application:

```
[HP][HP ODBC Driver] Communication link failure. The server timed out or disappeared
Platform: PC, Transport: TCPIP, Api: SQLFETCH, Error type: DRIVER,
Process: TCP:16.107.156.160/18655:NonStopODBC, Operation: DO_WRITE_READ,
function: RECV, error: 10054, error_detail: 0.
```

If you cancel a waiting query from an ODBC or JDBC client application, including NCI, WMS returns this message to the client application:

```
Query Canceled-Query Canceled By User.
```

A WMS Commands

Use these commands to configure WMS:

- "ADD SERVICE Command" (page 56)
- "ALTER SERVICE Command" (page 58)
- "ALTER WMS Command" (page 60)
- "DELETE SERVICE Command" (page 62)

Use these commands to display status information about queries, services, WMS or system resources:

- "INFO SERVICE Command" (page 64)
- "INFO WMS Command" (page 65)
- "STATUS Command" (page 66)

Use these commands to manage WMS workload:

- "CANCEL Command" (page 61)
- "HOLD Command" (page 63)
- "RELEASE Command" (page 66)

How to Execute WMS Commands in NCI

By default, NCI sessions are in SQL mode. To execute WMS commands in NCI, you must be in NS mode.

To change to NS mode, enter the MODE NS command. After you enter the MODE NS command in NCI, the prompt changes to NS% as this example shows:

```
SQL>mode ns

NS%
```

You are now ready to execute WMS commands in NCI. Terminate each WMS command that you execute in NCI with a semicolon (;).

TIP: Consider putting WMS commands in script files that you can run in NCI. For more information about using script files in NCI, see the *Neoview Command Interface (NCI) Guide*.

ADD SERVICE Command

The ADD SERVICE command adds a service to the WMS configuration on the Neoview platform.

Syntax

```
ADD SERVICE service-name [service-attribute[, service-attribute]...]

service-attribute is:
    { priority | service-threshold | output-option | comment }

priority is:
    PRIORITY { URGENT | HIGH | MEDIUM-HIGH | MEDIUM | LOW-MEDIUM | LOW }

service-threshold is:
    { MAX_CPU_BUSY percentage }
  | { MAX_MEM_USAGE percentage }
  | { ACTIVE time-value TO time-value }
  | { EXEC_TIMEOUT query-timeout-value }
  | { WAIT_TIMEOUT query-timeout-value }

output-option is:
    { PLAN | NO_PLAN }
  | { TEXT | NO_TEXT }

comment is:
    { COMMENT "comment-string" }
```

service-name

is the name of a user-defined service for managing queries. *service-name* can be a maximum of 24 characters and must be a unique name in WMS. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

priority

is the priority of this service relative to other services. *priority* can be one of these values in order of descending priority:

- PRIORITY URGENT
- PRIORITY HIGH
- PRIORITY MEDIUM-HIGH
- PRIORITY MEDIUM
- PRIORITY LOW-MEDIUM
- PRIORITY LOW

If you do not specify a priority, the default is MEDIUM. For more information about setting priorities, see “How Service Priorities and System Resource Thresholds Work” (page 25).

service-threshold

specifies one of these limits for the service:

MAX_CPU_BUSY *percentage*	specifies the maximum CPU busy, as a percentage, of all the CPUs on the system for queries executing in the service. *percentage* can be a number from 0 through 100. Zero, the default value, means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE *percentage*	specifies the maximum memory usage, as a percentage, of the entire system for queries executing in the service. *percentage* can be a number from 0 through 100. Zero, the default value, means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.
ACTIVE *time-value* TO *time-value*	sets a start and end time range during which the service executes queries. *time-value* is *hh*:*mm*, where *hh* is a value for the hour and *mm* is a value for minutes. The active time cannot span more than one day. If not specified, the service is always active from 00:00 through 23:59.
EXEC_TIMEOUT *query-timeout-value*	specifies a timeout value for queries executing in the service. When the timeout value is reached, WMS cancels the executing query and stops the NDCS server. *query-timeout-value* can be a number from 0 through 1440 minutes. If not specified, the default is 0, meaning no timeout.
WAIT_TIMEOUT *query-timeout-value*	specifies a timeout value for queries waiting to execute in the service. When the timeout value is reached, WMS cancels the waiting query and returns an error message. *query-timeout-value* can be a number from 0 through 1440 minutes. If not specified, the default is 0, meaning no timeout.

output-option

specifies one of these options for storing information about the queries executing in the service:

PLAN	collects the execution plans of queries executing in the service so that client tools, such as NCI and the Neoview Manageability Repository, can use and display that data. If not specified, the default is NO_PLAN, meaning that no plans are collected.
NO_PLAN	does not collect the execution plans of queries executing in the service. The default is NO_PLAN.
TEXT	collects the SQL syntax of queries executing in the service so that client tools, such as NCI and the Neoview Manageability Repository, can use and display that data. If not specified, the default is NO_TEXT, meaning that no SQL text is collected.
NO_TEXT	does not collect the SQL syntax of queries executing in the service. The default is NO_TEXT.

comment

is optional text that you can specify to describe the service you are creating. *comment-string* cannot exceed 256 characters. In addition to alphanumeric characters, the comment can include dashes, periods, underscores, and spaces. You must delimit the comment in double quotes.

Considerations

- Use a unique name for the service. You cannot add a service that already exists.
- You cannot add a service named SYSTEM or HP_DEFAULT_SERVICE.
- To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command adds a service, DATALOADING, to the WMS configuration. The DATALOADING service has URGENT priority, allows a maximum CPU busy of 90 percent, and has an active period of 18:00 to 22:00:

  ```
  ADD SERVICE dataloading PRIORITY URGENT,
  MAX_CPU_BUSY 90, ACTIVE 18:00 TO 22:00
  ```

- This command gives the SALESQUERIES service MEDIUM priority and adds a comment that describes the service:

  ```
  ADD SERVICE salesqueries PRIORITY MEDIUM,
  COMMENT "manages sales queries"
  ```

ALTER SERVICE Command

The ALTER SERVICE command modifies the attributes of a service in the WMS configuration on the Neoview platform. Altering a service takes effect immediately for all subsequently executing queries associated with that service.

Syntax

```
ALTER SERVICE service-name [service-attribute[, service-attribute]...]

service-attribute is:
    { priority | service-threshold | output-option | comment }

priority is:
    PRIORITY { URGENT | HIGH | MEDIUM-HIGH | MEDIUM | LOW-MEDIUM | LOW }

service-threshold is:
    { MAX_CPU_BUSY percentage }
  | { MAX_MEM_USAGE percentage }
  | { ACTIVE time-value TO time-value }
  | { EXEC_TIMEOUT query-timeout-value }
  | { WAIT_TIMEOUT query-timeout-value }

output-option is:
    { PLAN | NO_PLAN }
  | { TEXT | NO_TEXT }

comment is:
    { COMMENT "comment-string" }
```

service-name

is the name of a user-defined service for managing queries. *service-name* must exist in the WMS configuration but cannot be the SYSTEM service. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

priority

is the priority of this service relative to other services. *priority* can be one of these values in order of descending priority:

- PRIORITY URGENT
- PRIORITY HIGH
- PRIORITY MEDIUM-HIGH

- PRIORITY MEDIUM
- PRIORITY LOW-MEDIUM
- PRIORITY LOW

If you do not specify a priority, the default is MEDIUM.

service-threshold specifies one of these limits for the service:

MAX_CPU_BUSY *percentage*	specifies the maximum CPU busy, as a percentage, of all the CPUs on the system for queries executing in the service. *percentage* can be a number from 0 through 100. Zero, the default value, means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE *percentage*	specifies the maximum memory usage, as a percentage, of the entire system for queries executing in the service. *percentage* can be a number from 0 through 100. Zero, the default value, means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.
ACTIVE *time-value* TO *time-value*	sets a start and end time range during which the service executes queries. *time-value* is *hh*:*mm*, where *hh* is a value for the hour and *mm* is a value for minutes. The active time cannot span more than one day. If not specified, the service is always active from 00:00 through 23:59.
EXEC_TIMEOUT *query-timeout-value*	specifies a timeout value for queries executing in the service. This timeout value affects only new incoming queries associated with the service and not queries that are already executing in the service. When the timeout value is reached, WMS cancels the executing query and stops the NDCS server. *query-timeout-value* can be a number from 0 through 1440 minutes. The default is 0, meaning no timeout.
WAIT_TIMEOUT *query-timeout-value*	specifies a timeout value for queries waiting to execute in the service. This timeout value affects only new incoming queries associated with the service and not queries that are already waiting in the service. When the timeout value is reached, WMS cancels the waiting query and returns an error message. *query-timeout-value* can be a number from 0 through 1440 minutes. The default is 0, meaning no timeout.

output-option specifies one of these options for storing information about the queries executing in the service:

PLAN	collects the execution plans of queries executing in the service so that client tools, such as NCI and the Neoview Manageability Repository, can use and display that data. If not specified, the default is NO_PLAN, meaning that no plans are collected.
NO_PLAN	does not collect the execution plans of queries executing in the service. The default is NO_PLAN.

TEXT	collects the SQL syntax of queries executing in the service so that client tools, such as NCI and the Neoview Manageability Repository, can use and display that data. If not specified, the default is NO_TEXT, meaning that no SQL text is collected.
NO_TEXT	does not collect the SQL syntax of queries executing in the service. The default is NO_TEXT.

`comment`

is optional text that you can specify to describe the service that you are altering. *`comment-string`* cannot exceed 256 characters. In addition to alphanumeric characters, the comment can include dashes, periods, underscores, and spaces. You must delimit the comment in double quotes.

Considerations

- Altering a service might affect existing queries associated with the service. For more information, see "How an Altered Service Affects Queries" (page 31).
- You cannot alter the SYSTEM service with the ALTER SERVICE command. Instead, use the ALTER WMS command. For more information, see "ALTER WMS Command" (page 60).
- You cannot alter the priority or active time of HP_DEFAULT_SERVICE.
- To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command alters the DATALOADING service to have MEDIUM-HIGH priority:

  ```
  ALTER SERVICE dataloading PRIORITY MEDIUM-HIGH
  ```

- This command alters the SALESQUERIES service to have a maximum CPU busy of 80 percent and a maximum memory usage of 70 percent:

  ```
  ALTER SERVICE salesqueries MAX_CPU_BUSY 80, MAX_MEM_USAGE 70
  ```

ALTER WMS Command

The ALTER WMS command modifies the thresholds of the WMS system configuration (that is, the SYSTEM service) on the Neoview platform.

Syntax

```
ALTER WMS system-threshold[, system-threshold]...

system-threshold is:
    { MAX_CPU_BUSY percentage }
  | { MAX_MEM_USAGE percentage }
  | { STATS_INTERVAL time-value }
```

system-threshold specifies one of these limits for the system:

MAX_CPU_BUSY *percentage*	specifies the maximum CPU busy, as a percentage, for all the CPUs on the system. *percentage* can be a number from 0 through 100. Zero, the default value, means that WMS uses the internal value of 100 percent.
MAX_MEM_USAGE *percentage*	sets the maximum memory usage, as a percentage, for the entire system. *percentage* can be a number from 0 through 100. Zero, the default value, means that WMS uses the internal value of 85 percent.
STATS_INVERVAL *time-value*	specifies the interval, in seconds, at which WMS collects runtime statistics for all executing queries managed by the WMS. *time-value* is a number from 5 through 300. If not specified, the interval is five seconds by default.

Considerations

To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command alters the system configuration to have a statistics refresh interval of 10 seconds:

  ```
  ALTER WMS STATS_INTERVAL 10
  ```

- This command alters the system configuration to have a maximum CPU busy of 80% and a maximum memory usage of 90%:

  ```
  ALTER WMS MAX_CPU_BUSY 80, MAX_MEM_USAGE 90
  ```

CANCEL Command

The CANCEL command cancels a query or a set of queries being managed by WMS on the Neoview platform.

Syntax

```
CANCEL type

type is:

    { QUERIES service-name [ ALL | state ]}
  | { QUERIES { ALL | state } }
  | { QUERY query-id }

state is:
    { EXECUTING | WAITING | HOLDING }
```

type specifies one of these options for cancelling queries:

QUERIES *service-name* [ALL]	cancels all queries in the specified service. *service-name* is the name of a user-defined service for managing queries. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.
QUERIES *service-name state*	cancels all queries in the specified query state in the specified service. *service-name* is the name of a user-defined service for managing queries. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*. *state* is one of these query states: • EXECUTING • WAITING • HOLDING
QUERIES ALL	cancels all queries.
QUERIES *state*	cancels all queries in the specified query state. *state* is one of these query states: • EXECUTING • WAITING • HOLDING
QUERY *query-id*	cancels the specified query. *query-id* is the query identifier and can be a maximum of 160 characters.

Considerations

To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command cancels all queries in the DATALOADING service:

  ```
  CANCEL QUERIES dataloading
  ```

- This command cancels all queries in the executing state in the SALESQUERIES service:

  ```
  CANCEL QUERIES salesqueries EXECUTING
  ```

- This command cancels all queries on the Neoview platform:

  ```
  CANCEL QUERIES ALL
  ```

- This command cancels all queries in the waiting state:

  ```
  CANCEL QUERIES WAITING
  ```

- This command cancels a specific query:

  ```
  CANCEL QUERY MXID0100100050121203677536...
  ```

DELETE SERVICE Command

The DELETE SERVICE command removes a service from the WMS configuration on the Neoview platform.

Syntax

```
DELETE SERVICE service-name [IMMEDIATE]
```

`service-name` is the name of a user-defined service for managing queries. `service-name` must exist in the WMS configuration but cannot be the SYSTEM service or the HP_DEFAULT_SERVICE. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

IMMEDIATE removes the service immediately from WMS, causing all queries in the service to be immediately cancelled. When IMMEDIATE is not specified, new queries are not allowed to enter the service, and any waiting queries are cancelled, but any executing queries are allowed to finish before the service is deleted.

Considerations

- You cannot delete SYSTEM or HP_DEFAULT_SERVICE.
- To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command removes the DATALOADING service from the WMS configuration:

  ```
  DELETE SERVICE dataloading
  ```

- This command removes the SALESQUERIES service immediately from the WMS configuration and cancels all queries associated with that service, including executing queries:

  ```
  DELETE SERVICE salesqueries IMMEDIATE
  ```

HOLD Command

The HOLD command places a waiting query or a set of waiting queries being managed by WMS on hold. You cannot hold an executing query. To cancel an executing query, see the "CANCEL Command" (page 61).

Syntax

```
HOLD type

type is:

    { WMS }
  | { SERVICE { service-name | ALL }
  | { QUERY  { query-id | ALL } }
```

`type` specifies one of these options for holding queries:

WMS	suspends all WMS activities and puts WMS in a holding state.
SERVICE *`service-name`*	holds all waiting queries in the specified service. *`service-name`* is the name of a user-defined service for managing queries. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.
SERVICE ALL	holds all services on the Neoview platform.
QUERY *`query-id`*	holds the specified waiting query. *`query-id`* is the query identifier and can be a maximum of 160 characters.
QUERY ALL	holds all waiting queries on the Neoview platform.

Considerations

To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command suspends all WMS activities and puts WMS in a holding state:

  ```
  HOLD WMS
  ```
- This command holds all waiting queries in the DATALOADING service:

  ```
  HOLD SERVICE dataloading
  ```
- This command holds all services on the Neoview platform:

  ```
  HOLD SERVICE ALL
  ```
- This command holds a specific waiting query:

  ```
  HOLD QUERY MXID01001000501212036775 36...
  ```
- This command holds all waiting queries on the Neoview platform:

  ```
  HOLD QUERY ALL
  ```

INFO SERVICE Command

The INFO SERVICE command returns the service name, priority, threshold limits, and any comments for one or all services in WMS.

Syntax

```
INFO SERVICE { ALL | service-name }
```

ALL
represents all services on the system.

service-name
is the name of a user-defined service for managing queries. *service-name* must exist in the WMS configuration. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.

Considerations

- The INFO SERVICE command returns this information about one service or all the services on the system:

Table A-1 INFO SERVICE Command Output

Column Name	Data Type	Description
SERVICE_NAME	VARCHAR(96)	Name of the service
SERVICE_PRIORITY	VARCHAR(10)	Priority of the service
MAX_CPU_BUSY	INT	Maximum CPU busy allowed for this service. If this value is zero, it means that the default value, means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE	INT	Maximum memory usage allowed for this service. If this value is zero, the default value, it means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.
ACTIVE_TIME	VARCHAR(19)	Time range during which the service executes queries, such as from 00:00 through 23:59

Table A-1 INFO SERVICE Command Output *(continued)*

Column Name	Data Type	Description
SQL_PLAN	VARCHAR(7)	PLAN indicates that the query execution plans are collected for display. NO_PLAN indicates that no plans are collected for display.
SQL_TEXT	VARCHAR(7)	TEXT indicates that the SQL syntax of executing queries are collected for display. NO_TEXT indicates that no SQL text is collected for display.
EXEC_TIMEOUT	INT	Timeout value for executing queries in the service
WAIT_TIMEOUT	INT	Timeout value for waiting queries in the service
COMMENT	VARCHAR(1024)	Optional comment that describes the service

- To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command displays information about the DATALOADING service:

  ```
  INFO SERVICE dataloading
  ```

- This command displays information about all the services on the system:

  ```
  INFO SERVICE ALL
  ```

INFO WMS Command

The INFO WMS command returns this information about the WMS system configuration:

- Maximum CPU busy
- Maximum memory usage
- Interval at which WMS collects query runtime statistics

Syntax

```
INFO WMS
```

Considerations

- The INFO WMS command returns this information about the WMS system configuration:

Table A-2 INFO WMS Command Output

Column Name	Data Type	Description
MAX_CPU_BUSY	INT	Maximum CPU busy allowed. If this value is zero, it means that WMS uses the internal value of 100 percent.
MAX_MEM_USAGE	INT	Maximum memory usage allowed. If this value is zero, it means that WMS uses the internal value of 85 percent.
STATS_INTERVAL	INT	Frequency of updates to query runtime statistics. The default value is five seconds.

- To execute this command in NS mode in NCI, see "How to Execute WMS Commands in NCI" (page 55).

Examples

- This command displays information about the WMS system configuration:

```
INFO WMS
```

RELEASE Command

The RELEASE command releases a query or a set of queries being managed by WMS from the holding state.

Syntax

```
RELEASE type

type is:

    { WMS }
  | { SERVICE { service-name | ALL } }
  | { QUERY { query-id | ALL } }
```

type specifies one of these options for releasing queries from the holding state:

WMS	releases WMS from the holding state.
SERVICE *service-name*	releases all queries in the specified service from the holding state. *service-name* is the name of a user-defined service for managing queries. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.
SERVICE ALL	releases all services on the Neoview platform from the holding state.
QUERY *query-id*	releases the specified query from the holding state. *query-id* is the query identifier and can be a maximum of 160 characters.
QUERY ALL	releases all queries on the Neoview platform from the holding state.

Considerations

To execute this command in NS mode in NCI, see “How to Execute WMS Commands in NCI” (page 55).

Examples

- This command releases WMS from the holding state:

  ```
  RELEASE WMS
  ```

- This command releases all queries in the DATALOADING service from the holding state:

  ```
  RELEASE SERVICE dataloading
  ```

- This command releases all services on the Neoview platform from the holding state:

  ```
  RELEASE SERVICE ALL
  ```

- This command releases a specific query from the holding state:

  ```
  RELEASE QUERY MXID0100100050121203677536...
  ```

- This command releases all queries on the Neoview platform from the holding state:

  ```
  RELEASE QUERY ALL
  ```

STATUS Command

The STATUS command returns information about the current status of WMS and the system resources, or the services or queries being managed by WMS.

Syntax

```
STATUS type

type is:

    { WMS [VERSION] }
  | { SERVICE [ ALL | service-name | STATS ] }
  | { QUERIES [ ALL [ STATS ]
                    | EXECUTING [STATS]
                    | WAITING
                    | HOLDING ] }
  | { QUERIES SERVICE service-name [EXECUTING | WAITING | HOLDING] }
  | { QUERIES ROLE {role-name | alias-name} [EXECUTING | WAITING | HOLDING] }
  | { QUERY ALL STATS
            | query-id [STATS | PLAN | TEXT] }
```

type specifies one of these items for which to return status information:

WMS	returns status information about WMS and system resources. For more information, see Table A-3 (page 68).
WMS VERSION	returns version information for the internal WMS processes. For more information, see Table A-4 (page 69) and Table A-5 (page 69).
SERVICE ALL	returns status information about all the services on the system. For more information, see Table A-6 (page 69).
SERVICE *service-name*	returns status information about the specified service. *service-name* is the name of a user-defined service for managing queries. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*.For more information, see Table A-6 (page 69)
SERVICE STATS	returns statistics for all the services on the system. For more information, see Table A-7 (page 69).
QUERIES ALL	returns status information about all the queries on the system. For more information, see Table A-8 (page 70).
QUERIES ALL STATS	returns runtime statistics information about all the queries on the system. For more information, see Table A-10 (page 71).
QUERIES *state*	returns status information about all the queries in the specified state. *state* is one of these query states: • EXECUTING • WAITING • HOLDING For more information, see Table A-8 (page 70).
QUERIES EXECUTING STATS	returns the total number of executing queries in various states on the system. For more information, see Table A-9 (page 71).
QUERIES SERVICE *service-name* [*state*]	returns status information about the queries that are associated with a service. For more information, see Table A-8 (page 70). *service-name* is the name of a user-defined service for managing queries. The service name is not case-sensitive and is always converted to uppercase. WMS supports multibyte characters in the service name. For guidelines on using multibyte character sets, see the *Neoview Character Sets Administrator's Guide*. *state* is one of these query states: • EXECUTING • WAITING • HOLDING If you do not specify a query state, the command returns status information for all queries associated with the service.

QUERIES ROLE {*role-name* | *alias-name*} [*state*]	returns status information about the queries that are associated with a role, such as ROLE.USER, or an alias of a role, such as dba1. If you specify a role, the STATUS command returns only the queries associated with the particular role and not the queries associated with the aliases of the role.For more information, see Table A-8 (page 70). *role-name* is the name of a user role, which can be a maximum of 17 characters. The role name is case-sensitive. *alias-name* is the name of an alias, which can be a maximum of 32 characters. The alias is case-sensitive. *state* is one of these query states: • EXECUTING • WAITING • HOLDING If you do not specify a query state, the command returns status information for all queries associated with the role or alias.
QUERY ALL STATS	returns runtime statistics information about all the queries on the system. For more information, see Table A-10 (page 71).
QUERY *query-id*	returns status information about a query. *query-id* is the query identifier and can be a maximum of 160 characters. For more information, see Table A-8 (page 70).
QUERY *query-id* STATS	returns runtime statistics of the specified query. *query-id* is the query identifier and can be a maximum of 160 characters. For more information, see Table A-10 (page 71).
QUERY *query-id* PLAN	returns the query execution plan of the specified query. *query-id* is the query identifier and can be a maximum of 160 characters. PLAN must be enabled for the service in which the query is running. For more information, see Table A-11 (page 73).
QUERY *query-id* TEXT	returns the SQL text of the specified query. *query-id* is the query identifier and can be a maximum of 160 characters. TEXT must be enabled for the service in which the query is running. For more information, see Table A-11 (page 73).

Considerations

- The STATUS WMS option returns this status information:

Table A-3 System Status Information

Column Name	Data Type	Description
STATE	VARCHAR(9)	State of the system, which is INIT (for initialization), ACTIVE, HOLD, SHUTDOWN, or STOP
MAX_CPU_BUSY	INT	Maximum CPU busy allowed. If this value is zero, it means that WMS uses the internal value of 100 percent.
MAX_MEM_USAGE	INT	Maximum memory usage allowed. If this value is zero, it means that WMS uses the internal value of 85 percent.
STATS_INTERVAL	INT	Interval at which WMS collects query runtime statistics
DISK_IO	FLOAT	Average disk I/O across the cluster
DISK_CACHE	FLOAT	Average disk cache usage across the cluster
CPU_BUSY	FLOAT	Average CPU busy across the cluster
MEM_USAGE	FLOAT	Average memory usage across the cluster

- The STATUS WMS VERSION option returns this version information:

Table A-4 System Version Information

Column Name	Data Type	Description
VERSION_INFO	VARCHAR(n)	A string of version information for internal WMS processes

Table A-5 VERSION_INFO Column String

String Element	Example
Segment number of the WMS process	01
Process name	QSSYNC
Component ID	33
Major version number	2
Minor version number	3
Build ID	1
VPROC information	T8010N26_15FEB2008_AFC_SYNC_0304

- The STATUS SERVICE option returns this status information:

Table A-6 Services Status Information

Column Name	Data Type	Description
SERVICE_NAME	VARCHAR(96)	Name of the service
STATE	VARCHAR(7)	State of the service, which is INIT (for initialization), ACTIVE, HOLD, or DELETING
SERVICE_PRIORITY	VARCHAR(10)	Priority of the service, which is URGENT, HIGH, MEDIUM-HIGH, MEDIUM, LOW-MEDIUM, or LOW
MAX_CPU_BUSY	INT	Maximum CPU busy allowed. If this value is zero, the default value, it means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 100%.
MAX_MEM_USAGE	INT	Maximum memory usage allowed. If this value is zero, the default value, it means that WMS uses the threshold of the WMS system configuration, if set, or the internal default threshold of 85%.
ACTIVE_TIME	VARCHAR(19)	Active time for this service, such as from 00:00 through 23:59
SQL_PLAN	VARCHAR(7)	PLAN indicates that the query execution plans are collected for display. NO_PLAN indicates that no plans are collected for display.
SQL_TEXT	VARCHAR(7)	TEXT indicates that the SQL syntax of executing queries are collected for display. NO_TEXT indicates that no SQL text is collected for display.
EXEC_TIMEOUT	INT	Timeout value for executing queries in the service
WAIT_TIMEOUT	INT	Timeout value for waiting queries in the service

- The STATUS SERVICE STATS option returns these statistics for all the services on the system:

Table A-7 Statistics for All Services on the System

Column Name	Data Type	Description
SEGMENT	INT	Segment number
SERVICE_NAME	VARCHAR(96)	Name of the service
STATE	VARCHAR(8)	State of the service, which is INIT (for initialization), ACTIVE, or HOLD
EXECUTING	INT	Number of executing queries for the service across the cluster

Table A-7 Statistics for All Services on the System *(continued)*

Column Name	Data Type	Description
WAITING	INT	Number of waiting queries for the service across the cluster
HOLDING	INT	Number of holding queries for the service across the cluster

- The STATUS QUERIES and STATUS QUERY without the STATS, PLAN, or TEXT option return this status information:

Table A-8 Query Status Information

Column Name	Data Type	Description
QUERY_ID	VARCHAR(160)	Query identifier
QUERY_STATE	VARCHAR(9)	State of the query, which is INIT, WAITING, EXECUTING, or HOLDING
SERVICE_NAME	VARCHAR(96)	Name of the service
EST_COST	DOUBLE PRECISION	Estimated cost associated with executing the query
EST_CPU_TIME	DOUBLE PRECISION	Estimated number of seconds of processor time it might take to execute the instructions of the query execution plan. A value of 1.0 is 1 second.
EST_IO_TIME	DOUBLE PRECISION	Estimated number of seconds of I/O time (seeks plus data transfer) to perform the I/O for this query
EST_MSG_TIME	DOUBLE PRECISION	Estimated number of seconds it takes for the messaging for the query. The estimate includes the time for the number of local and remote messages and the amount of data sent.
EST_IDLE_TIME	DOUBLE PRECISION	Estimated number of seconds to wait for an event to happen. Events include opening a table or starting an ESP process.
EST_TOTAL_TIME	DOUBLE PRECISION	Estimated number of seconds to execute the query
EST_CARDINALITY	DOUBLE PRECISION	Estimated number of rows that the query will return
PROCESS_NAME	VARCHAR(32)	Process name of the NDCS server (MXOSRVR)
QUERY_NAME	VARCHAR(200)	Application ID and computer name delimited by the || character
ROLE_NAME	VARCHAR(50)	Client role
SERVICE_INDEX	INT	Sequential number of the service
QUERY_INDEX	INT	Sequential number of the query
START_TS	VARCHAR(26)	Timestamp when the query was registered in WMS
ENTRY_TS	VARCHAR(26)	Timestamp when the query was inserted into the service
ENTRY_JTS	LARGEINT	Unformatted timestamp when the query was inserted into the service
LAST_UPDATED	LARGEINT	Local civil time when the shared memory block was last updated

- The STATUS QUERIES EXECUTING STATS returns these numbers of executing queries in various states on each segment of the Neoview platform:

Table A-9 Numbers of Executing Queries on the System

Column Name	Data Type	Description
SEGMENT	INT	Segment number
TOT_INITIAL	INT	Number of executing queries in the initialization state for this segment
TOT_OPEN	INT	Number of executing queries in the open state for this segment
TOT_EOF	INT	Number of executing queries in the end-of-file state for this segment
TOT_CLOSE	INT	Number of executing queries in the close state for this segment
TOT_DEALLOCATED	INT	Number of executing queries in the deallocate state for this segment
TOT_FETCH	INT	Number of executing queries in the fetch state for this segment
TOT_CLOSE_TABLES	INT	Number of executing queries in the close_tables state for this segment
TOT_PREPARE	INT	Number of executing queries in the prepare state for this segment

- The STATUS QUERY with STATS, STATUS QUERY ALL STATS, and STATUS QUERIES ALL STATS return this status information:

Table A-10 Query Runtime Statistics

Column Name	Data Type	Description
QUERY_ID	VARCHAR(160)	Query identifier
QUERY_STATE	VARCHAR(9)	State of the query, which is INIT (for initialization), WAITING, EXECUTING, or HOLDING
STATEMENT_ID	VARCHAR(160)	Statement identifier specified by the client application
STATEMENT_TYPE	VARCHAR(21)	— OTHER — UNKNOWN — SELECT_UNIQUE — SELECT_NON_UNIQUE — INSERT_UNIQUE — INSERT_NON_UNIQUE — UPDATE_UNIQUE — UPDATE_NON_UNIQUE — DELETE_UNIQUE — DELETE_NON_UNIQUE — CONTROL — SET_TRANSACTION — SET_CATALOG — SET_SCHEMA — CALL_NO_RESULT_SETS — CALL_WITH_RESULT_SETS — SP_RESULT_SET
COMP_START_TIME	LARGEINT	Compilation start time
COMP_END_TIME	LARGEINT	Compilation end time
EXEC_START_TIME	LARGEINT	Execution start time
EXEC_END_TIME	LARGEINT	Execution end time

Table A-10 Query Runtime Statistics *(continued)*

Column Name	Data Type	Description
EXEC_STATE	VARCHAR(21)	— INITIAL — OPEN — EOF — CLOSE — DEALLOCATED — FETCH — CLOSE_TABLES — PREPARE — PROCESS_ENDED — UNKNOWN
ELAPSED_TIME	LARGEINT	Elapsed time of query execution
ACCESSED_ROWS	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_ROWS	LARGEINT	Number of rows actually used by the query.
MESSAGE_COUNT	LARGEINT	Number of messages sent to execute operations on the tables referenced by the query, including messages sent by the file system to the disk process and messages sent by the executor to the compiler for recompilation. For example, a FETCH operation through a secondary index generally sends two messages.
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
LOCK_ESCALATIONS	LARGEINT	Number of times a row lock was escalated to a file lock for the query
PROCESS_BUSYTIME	LARGEINT	The actual CPU time in microseconds spent by all DP2s involved in executing the query
OPENS	LARGEINT	Number of OPEN calls performed by the executor on behalf of this process
OPEN_TIME	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
NUM_ROWS_IUD	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 RTS. A warning of 8922 means that one or more CPUs did not report the statistics for a given query.
SQL_SPACE_ALLOC	INT	Amount, in kilobytes, of the space type of memory allocated in the master and executor server processes (ESPs) involved in processing the query
SQL_SPACE_USED	INT	Amount, in kilobytes, of the space type of memory used in the master and ESPs involved in processing the query
SQL_HEAP_ALLOC	INT	Amount, in kilobytes, of the heap type of memory allocated in the master and ESPs involved in processing the query

Table A-10 Query Runtime Statistics *(continued)*

Column Name	Data Type	Description
SQL_HEAP_USED	INT	Amount, in kilobytes, of the heap type of memory used in the master and ESPs involved in processing the query
SQL_CPU_TIME	LARGEINT	Approximation, in microseconds, of the total CPU time spent in the master and ESPs involved in processing the query
EID_SPACE_ALLOC	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	INT	Amount, in kilobytes, of the space type of memory used in the EID involved in processing the query
EID_HEAP_ALLOC	INT	Amount, in kilobytes, of the heap type of memory allocated in the EID involved in processing the query
EID_HEAP_USED	INT	Amount, in kilobytes, of the heap type of memory used in the EID involved in processing the query
EST_ACCESSED_ROWS	FLOAT	Estimated number of rows accessed by the query
EST_USED_ROWS	FLOAT	Estimated number of rows used by the query
PROCESSES_CREATED	LARGEINT	Number of processes generated by the executor on behalf of this query
PROCESS_CREATE_TIME	LARGEINT	Time spent creating processes
LAST_UPDATED	LARGEINT	Local civil time when the shared memory block was last updated.

- The STATUS QUERY with PLAN returns the query execution plan:

Table A-11 Query Execution Plan

Column Name	Data Type	Description
QUERY_PLAN	VARCHAR(3800)	Text for the query execution plan, which is returned in ISO8859-1 encoding. To interpret query execution plans, see the *Neoview Query Guide*.

- The STATUS QUERY with TEXT returns the SQL syntax for the query:

Table A-12 SQL Text

Column Name	Data Type	Description
QUERY_TEXT	VARCHAR(3800)	Text for the SQL query syntax, which is returned in UTF8 encoding.

- To execute this command in NS mode in NCI, see “How to Execute WMS Commands in NCI” (page 55).

Examples

- This command displays status information about WMS:

```
STATUS WMS
```

- This command displays version information for the internal WMS processes:

```
STATUS WMS VERSION
```

- This command displays status information for all the services on the system:

```
STATUS SERVICE ALL
```

- This command displays status information for the SALESQUERIES service:

  ```
  STATUS SERVICE salesqueries
  ```

- This command displays statistics for all the services on the system:

  ```
  STATUS SERVICE STATS
  ```

- This command displays status information for all the queries on the system:

  ```
  STATUS QUERIES ALL
  ```

- This command displays runtime statistics information about all the queries on the system:

  ```
  STATUS QUERIES ALL STATS
  ```

- This command displays status information for all the waiting queries on the system:

  ```
  STATUS QUERIES WAITING
  ```

- This command displays the total number of executing queries in various states on the system:

  ```
  STATUS QUERIES EXECUTING STATS
  ```

- This command displays status information for all queries in the DATALOADING service:

  ```
  STATUS QUERIES SERVICE dataloading
  ```

- This command displays status information for the executing queries in the DATALOADING service:

  ```
  STATUS QUERIES SERVICE dataloading EXECUTING
  ```

- This command displays status information for all the queries that belong to the role, role.mgr:

  ```
  STATUS QUERIES ROLE role.mgr
  ```

- This command displays status information for the holding queries that belong to the role, role.mgr:

  ```
  STATUS QUERIES ROLE role.mgr HOLDING
  ```

- This command displays runtime statistics information about all the queries on the system:

  ```
  STATUS QUERY ALL STATS
  ```

- This command displays status information of the specified query:

  ```
  STATUS QUERY MXID01001000050121203677536...
  ```

- This command displays the runtime statistics of the specified query:

  ```
  STATUS QUERY MXID01001000050121203677536... STATS
  ```

- This command displays the query execution plan of the specified query:

  ```
  STATUS QUERY MXID01001000050121203677536... PLAN
  ```

- This command displays the SQL text of the specified query:

  ```
  STATUS QUERY MXID01001000050121203677536... TEXT
  ```

Index