Teradata Vantage™ - Teradata® Virtual Storage

Release 17.10

July 2021

B035-1179-171K
DOCS.TERADATA.COM

Copyright and Trademarks

Copyright © 2010 - 2021 by Teradata. All Rights Reserved.

All copyrights and trademarks used in Teradata documentation are the property of their respective owners. For more information, see Trademark Information.

Product Safety

Safety type	Description
NOTICE	Indicates a situation which, if not avoided, could result in damage to property, such as to equipment or data, but not related to personal injury.
⚠ CAUTION	Indicates a hazardous situation which, if not avoided, could result in minor or moderate personal injury.
⚠ WARNING	Indicates a hazardous situation which, if not avoided, could result in death or serious personal injury.

Third-Party Materials

Non-Teradata (i.e., third-party) sites, documents or communications ("Third-party Materials") may be accessed or accessible (e.g., linked or posted) in or in connection with a Teradata site, document or communication. Such Third-party Materials are provided for your convenience only and do not imply any endorsement of any third party by Teradata or any endorsement of Teradata by such third party. Teradata is not responsible for the accuracy of any content contained within such Third-party Materials, which are provided on an "AS IS" basis by Teradata. Such third party is solely and directly responsible for its sites, documents and communications and any harm they may cause you or others.

Warranty Disclaimer

Except as may be provided in a separate written agreement with Teradata or required by applicable law, the information available from the Teradata Documentation website or contained in Teradata information products is provided on an "as-is" basis, without warranty of any kind, either express or implied, including the implied warranties of merchantability, fitness for a particular purpose, or noninfringement.

The information available from the Teradata Documentation website or contained in Teradata information products may contain references or cross-references to features, functions, products, or services that are not announced or available in your country. Such references do not imply that Teradata Corporation intends to announce such features, functions, products, or services in your country. Please consult your local Teradata Corporation representative for those features, functions, products, or services available in your country.

The information available from the Teradata Documentation website or contained in Teradata information products may be changed or updated by Teradata at any time without notice. Teradata may also make changes in the products or services described in this information at any time without notice.

Machine-Assisted Translation

Certain materials on this website have been translated using machine-assisted translation software/tools. Machine-assisted translations of any materials into languages other than English are intended solely as a convenience to the non-English-reading users and are not legally binding. Anybody relying on such information does so at his or her own risk. No automated translation is perfect nor is it intended to replace human translators. Teradata does not make any promises, assurances, or guarantees as to the accuracy of the machine-assisted translations provided. Teradata accepts no responsibility and shall not be liable for any damage or issues that may result from using such translations. Users are reminded to use the English contents.

Feedback

To maintain the quality of our products and services, e-mail your comments on the accuracy, clarity, organization, and value of this document to: docs@teradata.com.

Any comments or materials (collectively referred to as "Feedback") sent to Teradata Corporation will be deemed nonconfidential. Without any payment or other obligation of any kind and without any restriction of any kind, Teradata and its affiliates are hereby free to (1) reproduce, distribute, provide access to, publish, transmit, publicly display, publicly perform, and create derivative works of, the Feedback, (2) use any ideas, concepts, know-how, and techniques contained in such Feedback for any purpose whatsoever, including developing, manufacturing, and marketing products and services incorporating the Feedback, and (3) authorize others to do any or all of the above.

Contents

Introduction to Teradata Virtual Storage

Introduction to Teradata Virtual Storage

Teradata Vantage™ is our flagship analytic platform offering, which evolved from our industry-leading Teradata® Database. Until references in content are updated to reflect this change, the term Teradata Database is synonymous with Teradata Vantage.

Teradata Virtual Storage (VS):

- Allocates storage
- Keeps track of where data is stored on the physical media
- Maintains statistics on the data temperature (frequency of access) and on the grade (performance metric of storage locations)
- Automatically migrates more frequently accessed data to faster physical storage and less frequently accessed data to slower physical storage

Changes and Additions

Date	Description
July 2021	Minor edits.

Benefits

Storage Optimization, Data Migration, and Data Evacuation

Mixing storage technologies (device types, speeds, capacities) in a parallel architecture typically requires significant user interaction to ensure that the most frequently accessed data is stored on the fastest storage. Teradata VS automates this process, transparently moving this important data to storage media that offers the fastest I/O performance.

Data temperature is an indication of the relative frequency of access to a quantity of data: frequently accessed data is considered to be "hot," and infrequently accessed data is considered to be "cold." Teradata VS maintains statistics on temperatures of cylinders, logical groups of row data, and also is aware of the relative data access speeds (storage *grades*) provided by of the various kinds of storage media . This allows Teradata VS to intelligently migrate more frequently accessed data to faster physical storage. As data access patterns change, Teradata VS can move cylinders to faster or slower physical storage within each clique to improve system performance over time.

Teradata VS maintains statistics on the data temperature and on the grade. This allows Teradata VS to intelligently migrate more frequently accessed data to faster physical storage. As data access patterns change, Teradata VS can move storage cylinders to faster or slower physical storage within each clique to improve system performance over time.

Teradata VS can migrate all data from one physical storage device to another to prepare for removal or replacement of the physical device. This process is called evacuation. Complete data evacuation requires a system restart, but Teradata VS supports a soft evacuation feature that allows much of the data to be migrated while the system remains online. This can minimize system down time when evacuations are necessary.

Lower Barriers to System Growth

Device management features of Teradata VS provide the ability to pool storage within each clique. Each storage device (pdisk) can be shared, if necessary, by all AMPs in a clique. If the number of storage devices is not a multiple of the number of AMPs in the clique, the extra storage is shared. Consequently, storage can be added to the system in smaller increments, as needs and opportunities arise.

Mixed Storage Support

Teradata VS supports the use of different disk types, capacities, and speeds used within cliques. *Hybrid storage* support (available with Enterprise Tier licensing) supports systems that have a mix of storage technologies, such as SSD and HDD storage, or write- and read-intensive SSDs. As new types of physical storage (such as new disk sizes and types) become available and are certified for use with Vantage, they can be added to any clique.

Considerations

Licensing

Teradata VS hybrid storage support is available with Vantage Enterprise Tier licensing. For more information, contact your Teradata sales representative.

Temperature-based block-level data compression (TBBLC) and temperature-based data migration between fast and slow storage media is available to all Vantage customers, regardless of licensing tier. For more information about TBBLC, see *Teradata Vantage™ - Database Design*, B035-1094 and *Teradata Vantage™ - Database Utilities*, B035-1102.

Initial Data Temperature

When data is loaded into a new table, the system does not know whether that data will be hot (accessed frequently) or cold (accessed infrequently), so it is initially assigned the temperature specified by the appropriate Storage setting in DBS Control. Typically, you can allow the system to set the temperature according to this setting.

Over time, Teradata VS monitors data access patterns, assigns table data a temperature based on those access patterns, and moves data between faster and slower storage to place the hotter data on faster physical storage.

For tables with very specific data access requirements, you can use the TVSTemperature query band (at load time) or the Ferret FORCE command (soon after a table is created) to expedite the time required to place the data into the proper storage grade.

Note:

Use the TVSTemperature query band and the Ferret FORCE command only if you are certain that the standard background migration will not move the data to appropriate storage quickly enough for your needs.

Teradata VS monitors frequency of data access and adjusts data temperatures dynamically as it monitors frequency of data access. Consequently, temperatures can change from those specified using the TVSTemperature query band or the FORCE command, if the measured frequency of actual data access differs from the temperature that is initially set.

2

Tuning and Troubleshooting Teradata Virtual Storage

This section provides information on using DBS Control and Ferret utilities to tune and troubleshoot Teradata VS. See *Teradata Vantage™ - Database Utilities*, B035-1102 for more information on working with the DBS Control and Ferret utilities.

The Virtual Storage collector and TVS Monitor portlet of Teradata Viewpoint allow you to collect and view statistics on data temperature and storage grade of cylinders allocated in Teradata VS. Statistics reflect current relationships between data temperature and storage grade, and historical trends in the management of storage grades based on data temperature. Temperature and grade metrics can help assess storage requirements for data, as well as aid in fine tuning the balance between disk performance and data-access demands. For more information about the TVS Monitor portlet, see *Teradata® Viewpoint User Guide*, B035-2206.

Teradata Virtual Allocation Manager (tvam program) is a tool for managing storage optimization and data migration aspects of Teradata VS. It should be used only under the direction of Teradata Support Center personnel. Documentation for tvam includes a man page and online help.

Tuning Teradata Virtual Storage

Some aspects of Teradata VS are controlled by these settings in the DBS Control utility:

- AutoCylPackColdData
- Storage Group
    - All Other Temperature
    - DEPOT Temperature
    - GLOBAL TEMP Fallback Temperature
    - GLOBAL TEMP Fallback CLOB Temperature
    - GLOBAL TEMP Primary Temperature
    - GLOBAL TEMP Primary CLOB Temperature
    - PERMANENT JOURNAL Temperature
    - PERM Fallback Temperature
    - PERM Fallback CLOB Temperature
    - PERM Primary Temperature
    - PERM Primary CLOB Temperature
    - SPOOL Temperature
    - WAL Temperature

AutoCylPackColdData

Determines whether AutoCylPack (automatic background cylinder packing) operations are performed on cylinders that store cold (infrequently accessed) data, in addition to cylinders storing more frequently accessed data.

Field Group

File System

Valid Settings

Setting	Description
TRUE	AutoCylPack processes cylinders holding cold data.
FALSE	AutoCylPack ignores cylinders holding cold data.

Default

FALSE

Changes Take Effect

After the DBS Control Record has been written.

Related Information

For more information on…	See…
DBS Control settings related to AutoCylPack and temperature-based block-level compression	DBS Control in *Teradata Vantage™ - Database Utilities*, B035-1102.
Free Space Percent	• FreeSpacePercent DBS Control field in *Teradata Vantage™ - Database Utilities*, B035-1102. • Ferret SHOWFSP command in *Teradata Vantage™ - Database Utilities*, B035-1102. • *Teradata Vantage™ - Database Administration*, B035-1093.

Storage Group

Each field in the Storage group specifies the initial temperature that will be assigned to data loaded or inserted into empty subtables of a particular type. The temperature represents the expected frequency of access to that data, and can determine whether the data is compressed, and where the data is stored.

Note:

Because the frequency of access is continually monitored by Vantage, the temperature can change over time from the initial temperature setting.

The individual Storage setting names and defaults are listed below.

Storage Field	Default Temperature
DEPOT Temperature	HOT
GLOBAL TEMP Fallback Temperature	WARM
GLOBAL TEMP Fallback CLOB Temperature	WARM
GLOBAL TEMP Primary Temperature	WARM
GLOBAL TEMP Primary CLOB Temperature	WARM
PERMANENT JOURNAL Temperature	HOT
PERM Fallback Temperature	WARM
PERM Fallback CLOB Temperature	WARM
PERM Primary Temperature The Parallel Upgrade Tool (PUT) sets the default for PERM primary data based on the system configuration. For some configurations the default is HOT.	WARM
PERM Primary CLOB Temperature	WARM
SPOOL Temperature	HOT
WAL Temperature	HOT
All Other Temperature Controls initial data temperature set for data loaded into empty subtables that are not specifically controlled by other Storage group fields.	WARM

Valid Settings

Setting	Description
HOT	Data is expected to be accessed frequently.
WARM	Data is expected to be accessed moderately frequently.
COLD	Data is expected to be accessed infrequently.

Setting	Description
DEFAULT	Uses the default temperature for this type of data. Resets the field to the Teradata default value for the specified table or data type.

Changes Take Effect

After the DBS Control record has been written.

Usage Notes

The Storage field settings are honored when data is loaded into empty tables by any of the following techniques:

- INSERT/INSERT … SELECT statement
- CREATE TABLE AS … WITH DATA statement
- CREATE JI statement
- Load utilities TPump, FastLoad, MultiLoad, Teradata Parallel Transporter
- Table Rebuild utility

Note:

The TVSTemperature query bands and the Ferret FORCE command override these storage settings.

Because data temperature is assigned at the cylinder level, after initial assignment, data in different partitions of a partitioned table can have different temperatures.

Data can be moved (*migrated*) automatically to faster or slower grades of storage as data temperatures change due to changing data access patterns.

Troubleshooting Teradata Virtual Storage

You can use the following Ferret commands to troubleshoot Teradata VS.

Command	Valid SCOPE Options	Function
SHOWCYLALLOC	Vprocs	Shows a summary of the number of allocated and free cylinders on the system categorized by the performance grade of their storage.
SHOWWHERE	Vprocs, tables, or WAL log	Displays information about cylinder allocation, grade, and temperature

SHOWCYLALLOC

Shows a summary of the number of allocated and free cylinders on the system, categorized by the performance grade of the storage media where they are stored.

If local storage for spool is configured, the LOCAL row is shown in the summary.

Syntax

```
{ SHOWCYLALLOC | SHOWC }
```

SHOWCYLALLOC Output

```
SYSTEM WIDE CYLINDER ALLOCATION SUMMARY
 +---------------+--------+------------+--------+----------------+
 |    STORAGE    | TOTAL  | ALLOCATED  |  FREE  |    FREE GBs    |
 +---------------+--------+------------+--------+----------------+
 | FAST RESERVED |    264 |        105 |   159  |       1.891271 |
 +---------------+--------+------------+--------+----------------+
 | FAST          |   1248 |        322 |   926  |      11.014570 |
 +---------------+--------+------------+--------+----------------+
 | MEDIUM        |   3688 |          2 |  3686  |      43.844173 |
 +---------------+--------+------------+--------+----------------+
 | SLOW          |   1292 |         22 |  1270  |      15.106376 |
 +---------------+--------+------------+--------+----------------+
 | LOCAL         |   1988 |        118 |  1870  |      22.243247 |
 +---------------+--------+------------+--------+----------------+
```

Related Information

The following Ferret commands are used with Teradata Virtual Storage, a storage product available from Teradata. For more information about these commands, see *Teradata Vantage™ - Teradata® Virtual Storage*, B035-1179:

- BEGIN OPTIMIZE
- CANCEL OPTIMIZE
- FORCE

SHOWWHERE

The SHOWWHERE command displays information about cylinder allocation and temperature.

Note:

The CreateFsysInfoTable and PopulateFsysInfoTable macros provide a way to generate SHOWWHERE-like file system information to an SQL table. The tdheatmap table function provides a frequency-of-access report for database objects on a per-AMP basis. For more information about file system information macros and functions, see *Teradata Vantage™ - SQL Operators and User-Defined Functions*, B035-1210.

Syntax

```
{ SHOWWHERE | SHOWW } [ /S | /M | /L ]
```

Syntax Elements

/S

Displays a summary listing of the cylinders showing one line for every cylinder type .

This is the default.

/M

Displays a medium length listing of the cylinders with one line for every cylinder type per AMP (vproc).

/L

Displays a long listing of the cylinders with one line for every cylinder type per AMP (vproc) per storage device.

Usage Notes

Display output is limited to cylinders in the current scope (one or more vprocs, tables, or the WAL log), defined with the SCOPE command. If no scope has been specified, SHOWWHERE shows information for all cylinders.

If temperature information is not available for the current scope, SHOWWHERE returns an error.

If local storage for spool is configured and there is at least one spool cylinder allocated in that storage class, LOCAL will display in the output lines for SPOOL and SPOOL POOL.

Example: Using SHOWWHERE to Display Cylinder Allocation and Temperature

```
Ferret  ==>
showwhere
SHOWWHERE result for Each AMP
           +---------------+------------+------------------+-------------------------+-------+
 Vproc     |     # of      |    Type    |                  |        Temperature      |       |
 Num       |     Cyls      |            |    Grade     %   |  %HOT   %WARM   %COLD   |  VH   |
           +---------------+------------+------------------+-------+-------+---------+-------+
           |          361* |       PERM |   FAST  |  100%  |    0% |    5% |    95%  |  100% |
           |            0* |       PERM | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |       PERM |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |           16  |        WAL |   FAST  |  100%  |    0% |   38% |    63%  |    0% |
           |            0  |        WAL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |        WAL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |           24  |      DEPOT |   FAST  |  100%  |    0% |    0% |   100%  |    0% |
           |            0  |      DEPOT | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |      DEPOT |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |          112  |   WAL POOL |   FAST  |  100%  |    0% |   93% |     7%  |    0% |
           |            0  |   WAL POOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |   WAL POOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            8* |      SPOOL |   FAST  |  100%  |  100% |    0% |     0%  |    0% |
           |            0* |      SPOOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |      SPOOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |          136* | SPOOL POOL |   FAST  |  100%  |  100% |    0% |     0%  |    0% |
           |            0* | SPOOL POOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* | SPOOL POOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            8* |GLOBAL TEMP |   FAST  |  100%  |    0% |    0% |   100%  |    0% |
           |            0* |GLOBAL TEMP | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |GLOBAL TEMP |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            8* |       JRNL |   FAST  |  100%  |    0% |    0% |   100%  |    0% |
           |            0* |       JRNL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |       JRNL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            8* |     PSPOOL |   FAST  |  100%  |  100% |    0% |     0%  |    0% |
           |            0* |     PSPOOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |     PSPOOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           +---------------+------------+------------------+-------+-------+---------+-------+
 TOTAL     |         3326  |            |   FAST  |  100%  |   28% |    6% |    66%  |   65% |
           |            0  |            | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |            |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           +---------------+------------+------------------+-------+-------+---------+-------+
* Reported in units of Large Cylinders
 (A Large Cylinder is 6 times the size of a small cylinder)


Ferret  ==>
showwhere /m


SHOWWHERE result for Each AMP
           +---------------+------------+------------------+-------------------------+-------+
 Vproc     |     # of      |    Type    |                  |        Temperature      |       |
 Num       |     Cyls      |            |    Grade     %   |  %HOT   %WARM   %COLD   |  VH   |
           +---------------+------------+------------------+-------+-------+---------+-------+
   0       |           45* |       PERM |   FAST  |  100%  |    0% |    2% |    98%  |  100% |
           |            0* |       PERM | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |       PERM |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            2  |        WAL |   FAST  |  100%  |    0% |   50% |    50%  |    0% |
           |            0  |        WAL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |        WAL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            3  |      DEPOT |   FAST  |  100%  |    0% |    0% |   100%  |    0% |
           |            0  |      DEPOT | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |      DEPOT |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |           14  |   WAL POOL |   FAST  |  100%  |    0% |   93% |     7%  |    0% |
           |            0  |   WAL POOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0  |   WAL POOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            1* |      SPOOL |   FAST  |  100%  |  100% |    0% |     0%  |    0% |
           |            0* |      SPOOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |      SPOOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |           17* | SPOOL POOL |   FAST  |  100%  |  100% |    0% |     0%  |    0% |
           |            0* | SPOOL POOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* | SPOOL POOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            1* |GLOBAL TEMP |   FAST  |  100%  |    0% |    0% |   100%  |    0% |
           |            0* |GLOBAL TEMP | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |GLOBAL TEMP |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            1* |       JRNL |   FAST  |  100%  |    0% |    0% |   100%  |    0% |
           |            0* |       JRNL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |       JRNL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           |            1* |     PSPOOL |   FAST  |  100%  |  100% |    0% |     0%  |    0% |
           |            0* |     PSPOOL | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
           |            0* |     PSPOOL |   SLOW  |    0%  |    0% |    0% |     0%  |    0% |
           +---------------+------------+------------------+-------+-------+---------+-------+
           +---------------+------------+------------------+-------+-------+---------+-------+
   1       |           44* |       PERM |   FAST  |  100%  |    0% |    5% |    95%  |  100% |
           |            0* |       PERM | MEDIUM  |    0%  |    0% |    0% |     0%  |    0% |
```

```
... and so forth, through:
           |            0* |     PSPOOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           +---------------+------------+-----------------+-------+-------+-------+-------+
           +---------------+------------+-----------------+-------+-------+-------+-------+
   7       |           44* |       PERM |   FAST    | 100%  |    2% |    2% |   95% |  100% |
           |            0* |       PERM |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |       PERM |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            2  |        WAL |   FAST    | 100%  |    0% |   50% |   50% |    0% |
           |            0  |        WAL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |        WAL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            3  |      DEPOT |   FAST    | 100%  |    0% |    0% |  100% |    0% |
           |            0  |      DEPOT |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |      DEPOT |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |           14  |   WAL POOL |   FAST    | 100%  |    0% |   93% |    7% |    0% |
           |            0  |   WAL POOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |   WAL POOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |      SPOOL |   FAST    | 100%  |  100% |    0% |    0% |    0% |
           |            0* |      SPOOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |      SPOOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |           17* | SPOOL POOL |   FAST    | 100%  |  100% |    0% |    0% |    0% |
           |            0* | SPOOL POOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* | SPOOL POOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |GLOBAL TEMP |   FAST    | 100%  |    0% |    0% |  100% |    0% |
           |            0* |GLOBAL TEMP |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |GLOBAL TEMP |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |       JRNL |   FAST    | 100%  |    0% |    0% |  100% |    0% |
           |            0* |       JRNL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |       JRNL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |     PSPOOL |   FAST    | 100%  |  100% |    0% |    0% |    0% |
           |            0* |     PSPOOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |     PSPOOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           +---------------+------------+-----------------+-------+-------+-------+-------+
           +---------------+------------+-----------------+-------+-------+-------+-------+
 TOTAL     |         3326  |            |   FAST    | 100%  |   28% |    6% |   66% |   65% |
           |            0  |            |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |            |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           +---------------+------------+-----------------+-------+-------+-------+-------+

* Reported in units of Large Cylinders
 (A Large Cylinder is 6 times the size of a small cylinder)

Ferret  ==>
showwhere /l


You have indicated that you want to do a long display for
the SHOWWHERE command.  Please be aware that the output can be
extremely large.

    Do you wish to continue?? (Y/N)
y

SHOWWHERE result for Each AMP
           +---------------+------------+-----------------+-----------------------+-------+
 Vproc     |     # of      |    Type    |                 |      Temperature      |       |
 Num  Disk |     Cyls      |            |   Grade     %   |  %HOT   %WARM  %COLD  |   VH  |
           +---------------+------------+-----------------+-------+-------+-------+-------+
   0   1   |           45* |       PERM |   FAST    | 100%  |    0% |    2% |   98% |  100% |
           |            0* |       PERM |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |       PERM |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            2  |        WAL |   FAST    | 100%  |    0% |   50% |   50% |    0% |
           |            0  |        WAL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |        WAL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            3  |      DEPOT |   FAST    | 100%  |    0% |    0% |  100% |    0% |
           |            0  |      DEPOT |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |      DEPOT |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |           14  |   WAL POOL |   FAST    | 100%  |    0% |   93% |    7% |    0% |
           |            0  |   WAL POOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |   WAL POOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |      SPOOL |   FAST    | 100%  |  100% |    0% |    0% |    0% |
           |            0* |      SPOOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |      SPOOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |           17* | SPOOL POOL |   FAST    | 100%  |  100% |    0% |    0% |    0% |
           |            0* | SPOOL POOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* | SPOOL POOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |GLOBAL TEMP |   FAST    | 100%  |    0% |    0% |  100% |    0% |
           |            0* |GLOBAL TEMP |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |GLOBAL TEMP |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |       JRNL |   FAST    | 100%  |    0% |    0% |  100% |    0% |
           |            0* |       JRNL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |       JRNL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           |            1* |     PSPOOL |   FAST    | 100%  |  100% |    0% |    0% |    0% |
           |            0* |     PSPOOL |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |     PSPOOL |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           +---------------+------------+-----------------+-------+-------+-------+-------+
 SUB-TOTAL |          415  |            |   FAST    | 100%  |   27% |    5% |   68% |   65% |
           |            0  |            |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0  |            |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
           +---------------+------------+-----------------+-------+-------+-------+-------+
   1   2   |           44* |       PERM |   FAST    | 100%  |    0% |    5% |   95% |  100% |
           |            0* |       PERM |  MEDIUM   |   0%  |    0% |    0% |    0% |    0% |
           |            0* |       PERM |   SLOW    |   0%  |    0% |    0% |    0% |    0% |
```

```
           |          2  |        WAL |   FAST    | 100%    |     0% |   50% |   50% |     0% |
... and so forth, through:
           +-------------+------------+-----------------+-------+-------+-------+-------+
  7   4    |         44* |       PERM |   FAST    | 100%    |     2% |    2% |   95% |   100% |
           |          0* |       PERM | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0* |       PERM |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |          2  |        WAL |   FAST    | 100%    |     0% |   50% |   50% |     0% |
           |          0  |        WAL | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0  |        WAL |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |          3  |      DEPOT |   FAST    | 100%    |     0% |    0% |  100% |     0% |
           |          0  |      DEPOT | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0  |      DEPOT |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |         14  |   WAL POOL |   FAST    | 100%    |     0% |   93% |    7% |     0% |
           |          0  |   WAL POOL | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0  |   WAL POOL |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |          1* |      SPOOL |   FAST    | 100%    |   100% |    0% |    0% |     0% |
           |          0* |      SPOOL | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0* |      SPOOL |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |         17* | SPOOL POOL |   FAST    | 100%    |   100% |    0% |    0% |     0% |
           |          0* | SPOOL POOL | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0* | SPOOL POOL |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |          1* |GLOBAL TEMP |   FAST    | 100%    |     0% |    0% |  100% |     0% |
           |          0* |GLOBAL TEMP | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0* |GLOBAL TEMP |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |          1* |       JRNL |   FAST    | 100%    |     0% |    0% |  100% |     0% |
           |          0* |       JRNL | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0* |       JRNL |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           |          1* |     PSPOOL |   FAST    | 100%    |   100% |    0% |    0% |     0% |
           |          0* |     PSPOOL | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0* |     PSPOOL |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           +-------------+------------+-----------------+-------+-------+-------+-------+
 SUB-TOTAL |        409  |            |   FAST    | 100%    |    29% |    5% |   66% |    65% |
           |          0  |            | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0  |            |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           +-------------+------------+-----------------+-------+-------+-------+-------+
 TOTAL     |       3326  |            |   FAST    | 100%    |    28% |    6% |   66% |    65% |
           |          0  |            | MEDIUM    |   0%    |     0% |    0% |    0% |     0% |
           |          0  |            |   SLOW    |   0%    |     0% |    0% |    0% |     0% |
           +-------------+------------+-----------------+-------+-------+-------+-------+


* Reported in units of Large Cylinders
  (A Large Cylinder is 6 times the size of a small cylinder)
```

3

Specifying Data Temperatures

Normally, Teradata VS monitors the frequency of data access at the cylinder level, and periodically migrates cylinders that store more frequently accessed (hotter) data to faster storage media, and less frequently accessed (colder) data to slower storage media. The monitoring process requires some time to gauge the frequency of access, before data can be migrated to appropriate media.

The temperature that is assigned to new data, before it has been migrated for the first time, is normally determined by defaults set in DBS Control. This section discusses techniques for explicitly specifying data temperatures, which can precede or override the normal monitoring and migration process.

For more information on the benefits and considerations of specifying and changing data temperatures, see Introduction to Teradata Virtual Storage.

TVSTemperature Query Bands

TVSTemperature query bands can be used to specify an initial temperature setting for subsets of table data. This allows the data to be stored on access-appropriate media sooner than the normal Teradata VS monitoring and migration process would otherwise provide. Use the TVSTemperature query bands when the frequency of access for different kinds of data is known or can be anticipated.

In most cases, the data will be assigned the temperature you specify, and will be moved more quickly to appropriate storage media. However, in cases where newly loaded table data is stored on existing cylinders that also store data from other tables, the temperature and storage location of those cylinders will not be changed.

The following TVSTemperature query bands can be used in any combination.

Query Band	Sets Initial Temperature For
TVSTEMPERATURE	Data for which a temperature is not specified by other TVSTemperature query bands. If this query band is not used, the initial temperature set for this data follows the defaults specified by the Storage settings in DBS Control.
TVSTEMPERATURE_ PRIMARY	Primary data for tables and secondary indexes
TVSTEMPERATURE_ PRIMARYCLOB	Primary data for LOBs that are eligible for compression.
TVSTEMPERATURE_ FALLBACK	Fallback data for tables and secondary indexes.
TVSTEMPERATURE_ FALLBACKCLOB	Fallback data for LOBs that are eligible for compression.

Each of the query bands can be set to either COLD, WARM, HOT, or VERYHOT.

Query Band Value	Set For Data Expected to be Accessed
COLD	Infrequently. Data is stored initially on relatively slow media.
WARM	Moderately frequently. Data is stored on media of moderate speed.
HOT	Very frequently. Data is stored initially on the fastest media.
VERYHOT	The most frequently accessed. Data is stored on the fastest available storage media. Vantage attempts to maintain this data in a memory cache for fastest access.

Note:

Although the data is initially stored on media that corresponds to the temperature set by the query bands, if subsequent monitoring by Vantage indicates the data is accessed more or less frequently than the set temperature, the data can be migrated to other media.

These query bands are used with load utilities such as FastLoad, MultiLoad, Teradata Parallel Data Pump. The following SQL statements, used to populate tables, support the temperatures that have been set using the TVSTemperature query bands:

- INSERT into an empty table
- INSERT ... SELECT into an empty table
- MERGE inserts into an empty table
- CREATE TABLE ... AS ... WITH DATA
- CREATE HASH INDEX
- CREATE JOIN INDEX

Example: Force Newly Loaded Historical Data to COLD

A customer is loading five years of historical data (not previously stored in the database) and intends to perform some background analytics. Given the age of the data and the relatively light usage being planned, this data should reside on slow storage intended for infrequently accessed data.

Without using the TVSTemperature query band, the newly loaded data would be stored on cylinders having the default temperature (specified in DBS Control), and subsequently migrated over time to cold storage, as the system determines that the data is infrequently accessed.

Using the query band to load this data directly to cold storage makes sense in this instance because the likely frequency of data access is clearly known.

Example: Forcing Newly Loaded Fallback Data to COLD

The following SQL statement causes the temperature for all fallback data loaded into an empty table during the current session to be classified as COLD, and stored on storage media (or portions of storage devices) that are graded as slow in relation to the other storage media.

```
SET QUERY_BAND = 'TVSTEMPERATURE_FALLBACK = COLD;'
FOR SESSION;
```

Example: Forcing Newly Loaded Primary Data to HOT

The following SQL statement causes the temperature for all primary data loaded into an empty table during the current session to be classified as HOT, and stored on relatively fast storage media. All other types of data loaded during this session will be classified as COLD.

```
SET QUERY_BAND = 'TVSTEMPERATURE_PRIMARY = HOT;
                  TVSTEMPERATURE = COLD;'
FOR SESSION;
```

Example: Forcing Newly Loaded Data to VERYHOT for Fastest Access

You can use query bands strategically to set the temperature of data being loaded into the system, as in the following example scenario.

The telephone company requires very fast response time on queries processing data from today and yesterday. However, the query volume varies greatly depending on the time of day:

- The volume is highest on the hour, every hour, from 7:00 a.m. to 6:00 p.m.
- Data loads occur every 10 minutes, and then a set of critical queries runs against the newly loaded data, and the data from today and yesterday.
- When an issue occurs, very fast processing is required on queries against this data.

Assume that there is enough physical memory on the system so that an entire load can fit into VERYHOT cache. The default setting for VERYHOT cache is 50% of FSG cache. If the customer has a 6-node system with 768 GB of memory per node, there would be more than 2 TB of VERYHOT cache available for this application.

To load the new data into VERYHOT:

```
SET QUERY_BAND = 'TVSTEMPERATURE = VERYHOT;'
FOR SESSION;
```

The older data will cool down on its own as access to that data becomes less frequent. The latest data is loaded as VERYHOT and will remaining VERYHOT until it cools down.

Related Information

For more information on ...	See ...
Data temperatures	*Teradata Vantage™ - Database Design*, B035-1094.
Query bands	• *Teradata Vantage™ - SQL Data Definition Language Detailed Topics*, B035-1184. • *Teradata Vantage™ - Database Administration*, B035-1093.

For more information on ...	See ...
Storage settings in DBS Control	*Teradata Vantage™ - Database Utilities*, B035-1102.
CREATE TABLE, CREATE JOIN INDEX, CREATE HASH INDEX statements	*Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
INSERT, INSERT ... SELECT, MERGE statements	*Teradata Vantage™ - SQL Data Manipulation Language*, B035-1146.

Ferret Commands

You can use the following Ferret utility commands to start an immediate Teradata VS migration, or explicitly set the data temperature of table rows or partitions at any time.

Command	Valid SCOPE Options	Function
BEGIN OPTIMIZE	None	Initiates immediate, high-priority migrations necessary to move cylinders modified by the FORCE command to appropriate classes of storage. Normal migration is a background, lower priority process.
CANCEL OPTIMIZE	None	Halts the immediate migration that was started by the BEGIN OPTIMIZE command, and returns migrations to a lower priority background process.
FORCE	Vprocs	Sets the classification of cylinders occupied by a table to a specified data temperature value, regardless of table data access history.

BEGIN OPTIMIZE

Initiates immediate, high-priority migrations necessary to move cylinders modified by the FORCE command to appropriate classes of storage. Normal migration is a background process with a lower priority.

Note:

BEGIN OPTIMIZE initiates optimization on all nodes and all tables of the system, not just the tables on which the FORCE command was recently used.

Optimization can cause extremely heavy loads on system resources, and have a negative impact on system performance. If BEGIN OPTIMIZE is used on a production system, it should be used only during scheduled maintenance windows.

Syntax

```
{ BEGIN OPTIMIZE | BEG OPT }
```

Usage Notes

Migrations resulting from BEGIN OPTIMIZE occur at the maximum possible speed, without regard to potential impacts to existing workloads. Because cylinders marked HOT or COLD using the FORCE command are at the maximum temperature extremes, these cylinders are likely to be the first selected for migration, depending on available space.

If space on storage media of appropriate performance grade is not available, some cylinders are moved out of FAST and SLOW storage locations to make room for the HOT and COLD table data, respectively.

Example: BEGIN OPTIMIZE

```
Ferret  ==>
> begin optimize

BEGIN OPTIMIZE command could consume significant system resources
        Do you wish to continue?? (Y/N)
> Y
Optimization has been started on all vprocs

If you wish to revert this optimization to a low level background mode, you can
issue a CANCEL OPTIMIZE command
```

Related Information

- CANCEL OPTIMIZE
- FORCE

- SHOWCYLALLOC

CANCEL OPTIMIZE

Halts the high-priority optimization data migration that was begun by the BEGIN OPTIMIZE command. Migration for the process of data storage optimization reverts to the normal lower-priority background process. The normal background migration process imposes a minimal workload on system resources.

Syntax

```
CANCEL { OPTIMIZE | OPT }
```

Example: CANCEL OPTIMIZE

```
Ferret  ==>
> cancel optimize

Optimization has been stopped on all vprocs
```

Related Information

- BEGIN OPTIMIZE
- FORCE
- SHOWCYLALLOC

FORCE

Sets the cylinders occupied by a table or range of rows, to a specified data temperature, regardless of how frequently the data has been historically accessed.

Note:

Use the FORCE command only under the direction of Teradata Support Center personnel.

This command can have short- and long-term performance effects on both the targeted table and on other tables. Forcing a specific temperature can impact performance for tables that were not explicitly referenced in the FORCE command.

Syntax

```
FORCE tid [ rowspec [ TO rowspec ] ]
  TEMPERATURE = { HOT | WARM | COLD | VERYHOT }
```

Syntax Elements

tid

Identifies the subtable for which cylinder temperatures are set.

A *tid* consists of a unique identifier for the table plus an identifier for one subtable component of the table, such as the primary data subtable. The table can be uniquely identified in either of two ways:

- Specify the name of the database to which the table belongs and the table name, separated by a period. The names individually or together must be delimited. The following formats are valid:
    - "*database_name*.*table_name*"
    - "*database_name*"."*table_name*"
    - '*database_name*.*table_name*'
    - '*database_name*'.'*table_name*'
- Specify the unique numeric identifier of the table, which consists of two numeric values separated by a space. The unique table identifier is the first two numeric values returned by the TABLEID command. These two numbers are common to all subtables that comprise the table.

  Note:

  The format of the input numbers depends on the current radix setting, which is displayed by the RADIX command.

The second part of the tid identifies the subtable, and is called a *type_and_index*. This value can be represented in a number of ways, which are explained in the documentation for the Ferret utility in *Teradata Vantage™ - Database Utilities*, B035-1102. The primary data subtable has a numeric *type_and_index* value of decimal 1024 or hex 400. Specifying a 0 causes the FORCE command to be applied to all subtables of the table.

***rowspec* [TO *rowspec*]**

- For a nonpartitioned NoPI table, the row or range of rows for which a data temperature is to be set.
- For a partitioned table, the "combined partition number" that identifies the partition for which a data temperature is to be set (or the combined partition numbers that identify a range of logically contiguous partitions).

  The combined partition number is a calculated value that takes into account any column partitions and the different levels of partitioning in multilevel partitioned tables. It uniquely identifies every partition in the table.

  For tables with only a single-level partition without column partitioning, the combined partition number for any row is the value of the system-derived PARTITION column of the row.

  For more information about calculating combined partition numbers for other types of tables, see *Teradata Vantage™ - Database Design*, B035-1094.

VERYHOT
HOT
WARM
COLD

The data temperature that will be set for the cylinders that store row data for the specified table or rows, regardless of the historical frequency of access for that data.

- VERYHOT cylinders are the most frequently accessed.
- HOT cylinders are frequently accessed.
- WARM cylinders are accessed with moderate frequency.
- COLD cylinders store the least frequently accessed data.

Usage Notes

For new tables, Vantage does not yet have metrics on data access to use in assigning a temperature to the table data, so relies on the defaults set in the Storage group of DBS Control settings. The FORCE command can be used to force a non-default temperature on new tables without waiting for the system to collect data access statistics.

For tables with definitions that include the BLOCKCOMPRESSION = AUTOTEMP option (that is, tables using the temperature-based block-level compression feature), forcing uncompressed data to colder temperatures can cause the data to be compressed, and forcing compressed data to warmer temperatures can cause the data to be uncompressed. If the target of the FORCE command is a large amount of data, this could impact performance in the short term.

In addition to setting the temperature for whole tables, FORCE can be used to set the temperature for a row, range of rows, or specific partitions. For example:

- When data in older rows of a nonpartitioned NoPI table is less frequently accessed than data in newer rows, you can use FORCE to set the older rows to COLD.
- When one or more partitions of a partitioned table are expected to be more frequently accessed than other partitions, you can use FORCE to set the frequently accessed partitions to HOT.

When temperature-based block-level compression is enabled, using FORCE to set infrequently accessed tables or partitions to COLD causes them to be compressed sooner than they otherwise would be.

Data temperature can change over time from the temperature specified with the FORCE command if the measured frequency of actual data access differs from the temperature set using the FORCE command.

The temperature of boundary cylinders that store rows from more than a single table are not affected by the FORCE command if COLD is specified.

The FORCE command ignores any table scope that have been previously specified, and operates only on the table that is explicitly specified in the FORCE command.

Example: Using FORCE to Change the Temperature of a Table

The following example changes the temperature of a table.

```
Ferret  ==>
> force 0 6CE 400 temperature = hot

FORCE command changed the temperature of table test.table1 to HOT.
The temperature of 117 cylinders have been changed to HOT.
```

Example: Using FORCE to Change the Temperature of a Row Partition

The following example sets the temperature for a subset of rows a partitioned table. In this case, the rows are those in the combined partition identified by an external partition number of 9.

```
Ferret  ==>
> force "test.ppi_t02" 400 9 temperature = hot

FORCE command changed the temperature of table test.ppi_t02
to HOT for the specified partition.
The temperature of 5 cylinders have been changed to HOT.
```

Related Information

For more information on…	See…
nonpartitioned NoPI tables	*Teradata Vantage™ - SQL Data Definition Language Syntax and Examples*, B035-1144.
PARTITION columns	*Teradata Vantage™ - Database Design*, B035-1094.

Resource Usage Table Columns for Teradata Virtual Storage

This section provides information on resource usage table columns used with Teradata VS.

For more information on resource usage, see *Teradata Vantage™ - Resource Usage Macros and Tables*, B035-1099.

About the Mode Column

The Mode column describes the kind of data reported by the resource usage column.

IF the Mode column is ...	THEN these resource usage columns ...
count	tally the number of times an event occurred, such as the number of disk reads or writes during the logging period.
max	report the maximum value recorded during the logging period. Most of the max columns have a max suffix in the column name (for example, IoRespMax).
min	report the minimum value recorded during the logging period. The min columns have a min suffix in the column name (for example, AvailableMin).
track	show the value of a countable item achieved at the end of the current logging period. An example of a countable item is a queue length.

ResUsageSpdsk Table Migration Columns

Migration columns, a subcategory of the ResUsageSpdsk table statistics, identify the number of cylinders that migrated to a different location on a device as well as the time, in centiseconds, of all migration I/Os used, incurred, or saved during the log period.

Note:

Each allocation is for a cylinder size worth of data, also known internally in the allocator as an extent. Thus the column names begin with *Ext* for extent.

Column Name	Mode	Description	Data Type
ExtMigrateFaster	count	Number of cylinders migrated to a faster location on a device. This count is for cylinders that were allocated on this device and migrated to a different location within the same device or migrated to a completely different device.	FLOAT

Column Name	Mode	Description	Data Type
		The following formula calculates a ExtMigrateSlower value, which is the number of cylinders migrated to slower locations: ExtMigrateSlower = ExMigrateTotal - ExMigrateFaster.	
ExtMigrateIOTimeBenefit	count	Estimates the total I/O time savings achieved by migrations completing in the log period. The I/O time savings include the improvement in response time caused by the new data arrangement up to the time horizon. ExtMigrateIOTimeBenefit does not include the cost of the migration I/Os and is a gross benefit, not a net benefit.	FLOAT
ExtMigrateIOTimeCost	count	Estimates the total cost, in centiseconds, incurred by migration I/Os completing during the log period, where *cost* is the extra time waited by all non-migration I/Os as a result of the migration I/O.	FLOAT
ExtMigrateIOTimeImprove	count	Estimates the percent improvement in average I/O response time due to migrations completing in the log interval. For example, if, right before a particular log interval, the average I/O response time was 10 milliseconds (ms), then the migration logs an ExtMigrateIOTimeImprove value of 10% in this interval. The average I/O response time after the log interval should be (100%-10%)*10ms = 9ms. Migration then logs an ExtMigrateIOTimeImprove of 1% in the next interval. The average I/O response time in the new log interval is (100%-1%)*9ms = 8.91ms. ExtMigrateIOTimeImprove is only an estimate. Its permanent improvement remains in effect as long as the workload does not change and newer migrations do not significantly alter the data arrangement. When the workload changes or new migrations affect data arrangement, response time changes in an unquantifiable way. Despite this, ExtMigrateIOTimeImprove is useful because it predicts actual system performance at least for short periods of time and can be used to understand why the migration algorithm is doing what it is doing.	FLOAT
ExtMigrateReadRespTot	count	Migration read I/O response time.	FLOAT
ExtMigrateWriteRespTot	count	Migration write I/O response time.	FLOAT
ExtMigrateTotal	count	Total number of cylinders migrated to a different physical location. For more information, see the ExtMigrateFaster field.	FLOAT

ResUsageSvdsk Table Migration Columns

Migration columns, a subcategory of the ResUsageSvdsk table statistics, identify the number of cylinders that migrated to a different location on a device as well as the time, in centiseconds, of all migration I/Os used, incurred, or saved during the log period.

Note:

Each allocation is for a cylinder size worth of data, also known internally in the allocator as an extent. Therefore, the column names begin with *Ext* for extent.

Column Name	Mode	Description	Data Type
ExtMigrateFaster	count	Number of cylinders migrated to faster locations (that is, migrations whose gross benefits are positive) for the associated AMP. The following formula calculates a Slower Migration value, which is the number of cylinders migrated to slower locations: SlowerMigration = ExtMigrateTotal - ExtMigrateFaster Cylinders are migrated to slower locations to make room for hotter cylinders to replace them.	FLOAT
ExtMigrateIOTimeCost	count	Estimates the total cost, in centiseconds, incurred by migration I/Os completing during the log period, where *cost* is the extra time waited by all non-migration I/Os as a result of the migration I/O.	FLOAT
ExtMigrateIOTimeBenefit	count	Estimates the total I/O time savings achieved by migrations completing in the log period. The I/O time savings include the improvement in response time caused by the new data arrangement up to the time horizon. This value does not include the cost of the migration I/Os and is a gross benefit, not a net benefit.	FLOAT
ExtMigrateIOTimeImprove	count	Estimates the percent improvement in average I/O response time due to migrations completing in the log interval. For example, if, right before a particular log interval, the averageI/O response time was 10 milliseconds (ms), then the migration logs an ExtMigrateIOTimeImprove value of 10% in this interval. The average I/O response time after the log interval should be (100%-10%)*10ms = 9ms. Migration then logs an ExtMigrateIOTimeImprove of 1% in the next interval. The average I/O response time in the new log interval is (100%-1%)*9ms = 8.91ms. ExtMigrateIOTimeImprove is only an estimate. Its permanent improvement remains in effect as long as the workload does not change and newer migrations do not significantly alter the data arrangement.	FLOAT

Column Name	Mode	Description	Data Type
		When the workload changes or new migrations affect data arrangement, response time changes in a nonquantifiable way. You can use this field to predict the actual system performance for short periods of time and to understand why the migration algorithm is doing what it is doing.	
ExtMigrateReadRespTot	count	Migration read I/O response time.	FLOAT
ExtMigrateWriteRespTot	count	Migration write I/O response time.	FLOAT
ExtMigrateTotal	count	Total number of cylinders migrated to a different physical location. For more information, see the ExtMigrateFaster column.	FLOAT

ResUsageSvpr Table Node Agent Columns

These columns identify the migration processing statistics reported by the Node Agent.

Column Name	Mode	Description	Data Type
NodeAgentMigrationsDone	count	Number of migration requests completed by the Node Agent.	FLOAT
NodeAgentMigrationsStarted	count	Number of migration requests started by the Node Agent.	FLOAT
NodeAgentStatProcessed	count	Number of statistics buffers processed by the Node Agent.	FLOAT

A

How to Read Syntax

This document uses the following syntax conventions.

Syntax Convention	Meaning
`KEYWORD`	Keyword. Spell exactly as shown. Many environments are case-insensitive. Syntax shows keywords in uppercase unless operating system restrictions require them to be lowercase or mixed-case.
variable	Variable. Replace with actual value.
number	String of one or more digits. Do not use commas in numbers with more than three digits. Example: 10045
`[ x ]`	`x` is optional.
`[ x | y ]`	You can specify `x`, `y`, or nothing.
`{ x | y }`	You must specify either `x` or `y`.
`x [...]`	You can repeat `x`, separating occurrences with spaces. Example: `x x x` See note after table.
`x [,...]`	You can repeat `x`, separating occurrences with commas. Example: `x, x, x` See note after table.
`x [`*delimiter*`...]`	You can repeat `x`, separating occurrences with specified delimiter. Examples: • If *delimiter* is semicolon: `x; x; x` • If *delimiter* is `{,|OR}`, you can do either of the following: ◦ `x, x, x` ◦ `x OR x OR x` See note after table.

Note:

You can repeat only the immediately preceding item. For example, if the syntax is:

```
KEYWORD x [...]
```

You can repeat x. Do not repeat KEYWORD.

If there is no white space between x and the delimiter, the repeatable item is x and the delimiter. For example, if the syntax is:

```
[ x, [...] ] y
```

- You can omit x: y
- You can specify x once: x, y
- You can repeat x and the delimiter: x, x, x, y

B

Starting the Utilities

Vantage offers several interfaces from which the utilities may be started and run.

Interface	Description
Database Window (DBW)	DBW is a graphical tool that connects to the Vantage console subsystem (CNS). CNS provides console services to utility programs that operate on the database level. Console utilities should be started from DBW. **Note:** Operators must be members of the tdtrusted user group to run console utilities, or must be logged in as root. Non-tdtrused users may be explicitly granted access to the console using the CNS GRANT command. For more information on the GRANT command, see the Database Window utility documentation in *Teradata Vantage™ - Database Utilities*, B035-1102. For low bandwidth connections, command-line interfaces to CNS are available, such as cnsterm and cnstool. Online documentation is available for cnsterm and cnstool in the form of Linux man pages. A subset of the console utilities can be run from the Remote Console portlet of Teradata Viewpoint. For more information, see *Teradata® Viewpoint User Guide*, B035-2206.
Linux command line	Utilities that run directly from the command line are primarily those that operate on the PDE level.

Titles of sections in *Teradata Vantage™ - Database Utilities*, B035-1102 that describe specific utilities reflect the utility common name followed by the name of the executable utility program enclosed in parentheses. For example, "Control GDO Editor (ctl)". Use the executable program name, in this example, `ctl`, to start the utility from the command line or Database Window.

Note:

Not all utilities support all available user interfaces. For a listing of supported user interfaces for a utility, see the documentation for that utility.

When started, some utilities present their own interactive command-line or graphical user interfaces. These utilities allow browsing and entering information, and continue running until they are explicitly stopped. Many utilities that present their own command environment are stopped by entering the QUIT command.

Some utilities that run from DBW can be stopped by issuing the `stop` *window_number* command from the DBW Supervisor window, where *window_number* is the numeric identifier of the DBW application window in which the utility is running.

Starting a Utility from Database Window

Database Window (DBW) is an X client program that requires an X server to be running on the local machine. DBW supports standard X Windows display forwarding. To ensure that the graphical user interface displays properly, you can use the standard `-display` option to specify the host name or IP address of the local machine.

To start a utility from Database Window:

1. If not already done, set up the database environment by typing:

   `tdatcmd`

   at the Linux command line.
2. Open DBW from the Linux command line by typing:

   `xdbw –display` *displayspec* `&`

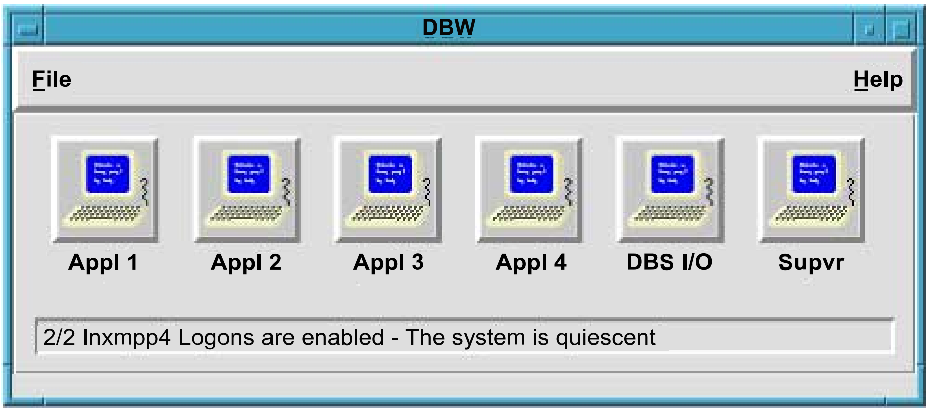
   where *displayspec* is the name or IP address of the local machine, followed by a colon and the server number, typically 0 or 0.0. For example:

   `xdbw -display myworkstation.mycompany.com:0.0 &`

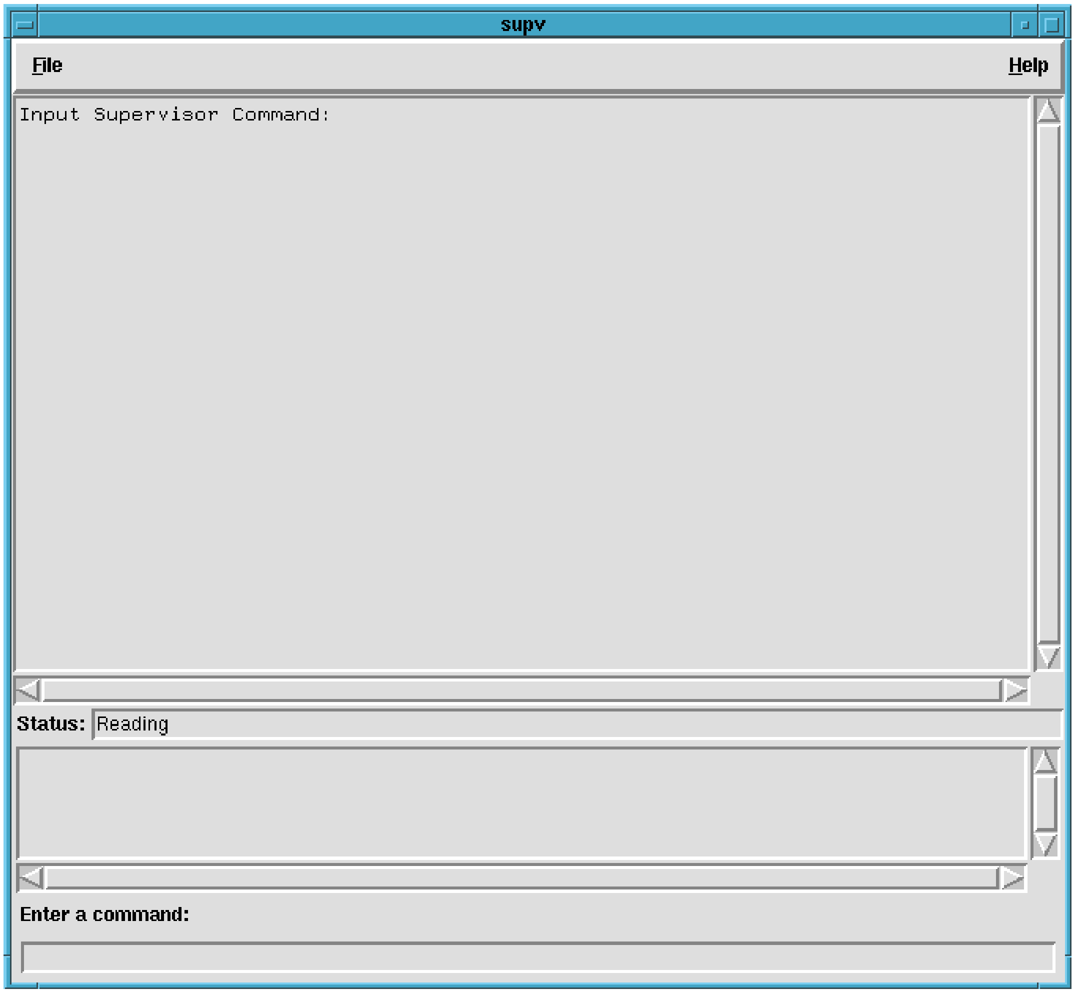
   or

   `xdbw -display 192.0.2.24:0.0 &`

   The **DBW** main window opens.



3. Click the **Supvr** button to open the **Supervisor** (**supv**) window.



4. Under **Enter a command**, type:

   `start` *utilityname* `[` *options* `]`

   where *utilityname* is the name of the utility, and *options* can include any of the available command-line options and arguments of that utility. *utilityname* is not case-sensitive.

   The following message appears:

   `Started '` *utilityname* `' in window` *x*

   where *x* is the number of one of the four available application windows DBW provides.

Note:

Utilities started from DBW run in the control vproc, usually vproc 0, unless the VPROC option of the `START` command is used to specify a different vproc. For more information, see the Database Window utility documentation in *Teradata Vantage™ - Database Utilities*, B035-1102.

Each utility started runs in one of the four application windows. The title bar of the application window and the corresponding button in the DBW main window change to reflect the name of the running utility. When the utility stops running, the application window and main window button revert to the default text title (that is, **Appl1**, **Appl2**, and so forth).

Note:

Up to four utilities can be run concurrently in DBW. The message `All Interactive Partitions are busy!!` indicates that all four application windows are occupied. In this case, you must quit from one of the four running utilities before another can be started.

For more information on DBW and CNS commands, and on options that are available with the `START` command, see Database Window in *Teradata Vantage™ - Database Utilities*, B035-1102.

Starting a Utility from the Linux Command Line

Note:

Operators must be members of the tdtrusted user group or have root access to run console utilities. Other users can run the utilities only if they have been explicitly granted access using the GRANT command. For more information on the GRANT command, see Database Window in *Teradata Vantage™ - Database Utilities*, B035-1102.

Some utilities cannot be run from the Linux command line. They must be run from DBW or an equivalent tool, such as cnsterm or cnstool.

To start a utility from the Linux command line:

1. If not already done, set up the database environment by typing:

   `tdatcmd`

   at the Linux command line.
2. On the command line type:

   utilityname [*options*]

   where *utilityname* is the name of the utility, and *options* can include any of the available command-line options and arguments of that utility.

Note:

Utilities started from the Linux command line run in the node vproc. For some operations, some utilities require that they be run on an AMP vproc. In these cases, run the utility from DBW or an equivalent tool.

Unicode support on the Linux command line requires a locale that supports UTF8 encoding. The output display screen must use a font that supports the required Unicode glyphs. For example, the following command sets an xterm window to use a common font that supports all Unicode glyphs:

```
xterm -fn '-efont-fixed-*-r-*-*-16-*-*-*-*-*-iso10646-1'
```

C

Additional Information

Teradata Links

Link	Description
https://docs.teradata.com/	Search Teradata Documentation, customize content to your needs, and download PDFs. Customers: Log in to access Orange Books.
https://support.teradata.com	One-stop source for Teradata community support, software downloads, and product information. Log in for customer access to: • Community support • Software updates • Knowledge articles
https://www.teradata.com/University/Overview	Teradata education network
https://support.teradata.com/community	Link to Teradata community